



**Universidade de Aveiro**  
2017

Departamento de Eletrónica, Telecomunicações e  
Informática

**Álvaro Rafael  
Moreira Santos**

**Plataforma de Medida com Suporte de Mobilidade**





**Universidade de Aveiro**  
**2017**

Departamento de Eletrónica, Telecomunicações e  
Informática.

**Álvaro Rafael  
Moreira Santos**

## **Plataforma de Medida com Suporte de Mobilidade**

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Rui Aguiar, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.



**Dedicatória**

Dedico este trabalho aos meus pais e restante família pelo incondicional e incansável apoio ao longo de todo este projeto.



## **o júri**

presidente

**Professor Doutor Paulo Bacelar Reis Pedreiras**  
Professor Auxiliar, Universidade de Aveiro

vogais

**Professor Doutor Jorge Augusto Fernandes Ferreira**  
Professor Auxiliar, Universidade de Aveiro

**Professor Doutor Rui Luís Andrade Aguiar**  
Professor Catedrático, Universidade de Aveiro (Orientador)





## **agradecimentos**

Agradecer primeiramente ao meu orientador por todo o tempo disponibilizado e esclarecimentos prestados sempre que necessário. De seguida, para além dos meus pais um também sincero obrigado a todos os meus amigos que funcionaram como pedras basilares neste percurso longo e árduo.



## palavras-chave

Mobilidade, AMazING, Carril, Posição, NodeMCU, BreakBeam

## resumo

A mobilidade foi sempre um fator-chave a ser suportado nas redes de testes sem fios (também conhecidas por *testbeds*) já existentes. Estas redes permitem a recriação de cenários reais e emulados, encontrando-se uma destas instalada no terraço do Instituto de Telecomunicações (IT) de Aveiro, de nome AMazING. Um carril metálico apoiado num total de 40 pilares, também eles metálicos, foi implementado no terraço do IT. Sobre este carril é colocado um protótipo derivado a partir de um motor de uma bicicleta elétrica capaz de se deslocar ao longo deste carril a velocidades variáveis, permitindo assim que se recriem ou testem cenários de mobilidade nesta rede. Ao longo deste documento diferentes estratégias e técnicas foram estudadas de acordo com o estado de arte, e é apresentada uma proposta de um sistema de localização de modo a estimar a posição deste protótipo, conforme o seu movimento ao longo do carril.



**keywords**

Mobility, AMazING, Rail, Position, NodeMCU, BreakBeam

**abstract**

Mobility has always been a key-factor to be inserted on already existent wireless networks designed for testing (also named testbeds). These networks allow the recreation of real and emulated scenarios and one of these is currently installed on the rooftop of the Institute of Telecommunications (IT) of Aveiro, named AMazING. A metallic rail supported by a total of 40 pillars, also metallic, has been deployed on the rooftop of IT. Over this rail is placed a prototype, based on the motor of an electrical bicycle, which is able of moving along the full length of the rail at variable values of velocity. This allows the recreation or testing of mobility scenarios on the testbeds. Throughout this document different strategies and techniques were studied accordingly to the state of the art and a proposal is presented of a localization system allowing this way to evaluate the position of the prototype, while it is travelling the metallic rail.



# ÍNDICE

<b>Lista de Figuras .....</b>	<b>iii</b>
<b>Lista de Tabelas .....</b>	<b>v</b>
<b>1 Introdução .....</b>	<b>1</b>
1.1 Motivação/Enquadramento .....	2
1.2 Objetivos do Trabalho .....	4
1.3 Estrutura do Documento .....	4
<b>2 Estado da Arte.....</b>	<b>6</b>
2.1 Uma Breve Introdução Histórica. ....	6
2.2 Localização – O Que É? A Chegada De Um Novo Sistema. ....	7
2.3 Um Outro Sistema De Localização. A Identificação Rádio Frequência .....	10
2.4 Localização Em Robótica .....	17
2.4.1 Localização Relativa / Local .....	17
2.4.2 Localização Global / Absoluta .....	19
2.4.2.1 Faróis Ativos.....	19
2.4.2.2 Tecnologia GPS E O RTK .....	20
2.4.2.3 Sistemas Ultrassónicos.....	24
2.4.2.4 Navegação Baseada Em Marcos .....	30
2.4.2.5 Navegação Baseada Em Mapas ("Map-Matching").....	32
<b>3 Sensores e o Seu Papel na Localização .....</b>	<b>35</b>
3.1 Sensor Ou Transdutor.....	35
3.2 Categorização E A Forma Como Sentem O Meio Ambiente.....	36
3.3 Sensores Inerciais .....	37
3.3.1 Acelerómetros .....	37
3.3.2 Giroscópios .....	39
3.4 Encoders (Codificadores).....	43
3.4.1 Encoders Óticos Incrementais .....	43
3.4.2 Encoders Óticos Absolutos .....	45
3.5 Sensores PIR e IR BreakBeam .....	46
<b>4 Solução Proposta .....</b>	<b>51</b>
4.1 Estrutura e Ideia .....	51
4.2 Implementação da Ideia .....	56
<b>5 Testes e Resultados .....</b>	<b>62</b>

<b>6</b>	<b>Conclusões .....</b>	<b>74</b>
6.1	Trabalho Futuro.....	75
	<b>Bibliografia.....</b>	<b>76</b>
	<b>Anexo A – Esquema Elétrico do Protótipo .....</b>	<b>80</b>
	<b>Anexo B – Software (Lado Cliente) .....</b>	<b>81</b>
	<b>Anexo C – Software (Lado Servidor) .....</b>	<b>83</b>



# LISTA DE FIGURAS

1. Esquemático da rede de testes <i>AMaZING</i> .....	3
2. Chegada ao Brasil por parte da armada Portuguesa [1] .....	6
3. Astrolábio [71] .....	6
4. Os diferentes sistemas GNSS e respetivos logotipos [2] .....	7
5. Segmentos de um sistema GNSS [3] .....	8
6. Erro devido ao fenómeno de <i>multipath</i> [4] .....	9
7. Um simples sistema RFID .....	10
8. Exemplar de um <i>Reader Ethernet</i> [5] .....	11
9. Exemplar de um <i>Reader WiFi</i> [6] .....	11
10. Exemplo de um <i>RFID Reader Handheld</i> [7] .....	12
11. <i>RFID Reader</i> Integrado [8] .....	12
12. <i>Metal Tag</i> [9] .....	13
13. <i>Wet Tag</i> [10] .....	13
14. Polarização circular [72] .....	14
15. Polarização linear [72] .....	14
16. Diagrama de um sistema RTK [45] .....	20
17. Ilustração de um sistema RTK utilizando os módulos NEO-M8P [11] .....	21
18. <i>Board Tiny RTK</i> da <i>drotek eletronic (Base station)</i> [45] .....	22
19. Sistema <i>Tiny RTK</i> da <i>drotek Eletronic</i> [45] .....	23
20. Sistema Ultrassónico da <i>IS Robotics</i> [19] .....	24
21. Medição dos três ângulos necessários para se obter triangulação [19] .....	25
22. Demonstração de funcionamento do sistema de navegação <i>TRC Beacon Navigation System</i> [19] .....	26
23. Exemplo de um <i>STROAB</i> [19] .....	27
24. Exemplar de um <i>NOAD</i> [19] .....	27
25. Sistema <i>CONAC™</i> [19] .....	28
26. Exemplar de um farol do sistema desenvolvido por [12] .....	29
27. Procedimento normal para navegação por marcos .....	30
28. Procedimento geral para navegação baseada em mapas .....	32
29. Princípio de funcionamento de um acelerómetro mecânico [13] .....	38
30. Acelerómetro Capacitivo [13] .....	38
31. Acelerómetro Piezoelétrico [13] .....	39
32. Giroscópio mecânico[14] .....	40
33. Efeito de Sagnac num anel ótico de raio R [15] .....	41
34. Giroscópio Anel Laser (RLG) [16] .....	41
35. Giroscópio fibra ótica (FOG) [14] .....	41
36. <i>Encoder Incremental</i> [17] .....	44
37. <i>Encoder incremental</i> e respetivo sinal de saída [18] .....	45
38. <i>Encoder Absoluto</i> com dois tipos de codificação [19] .....	46
39. Sensor Passivo infravermelho. Vista traseira e frontal, respetivamente [20] .....	47

40. Processo de detecção de um corpo quente por parte de um sensor PIR [20] .....	47
41. Sensores Infravermelhos Quebra Feixe [21] .....	49
42. Estrutura montada no terraço do IT .....	52
43. Veículo escolhido como carga .....	53
44. Caixa de controlo do veículo .....	53
45. Esquema simples do método de detecção de pilares por parte do sistema .....	54
46. Pilar de referência (a azul ciano) e mapeamento dos pilares .....	55
47. <i>NodeMCU devKit board</i> e a disposição dos diferentes pinos da <i>board</i> .....	56
48. Alimentação do <i>NodeMCU</i> (lado Carga) e dos Sensores <i>BreakBeam</i> .....	58
49. Comunicação entre Cliente e Servidor .....	59
50. Fluxograma do software implementado .....	60
51. Estruturas colocadas de modo a identificar o pilar de referência .....	63
52. Pilar referência e restantes pilares vizinhos .....	63
53. Perfil de uma reta do carril (Pilares número 7 a 13) .....	64
54. Perfil da curva que engloba pilares número 14 a 16 .....	64
55. Montagem do nó móvel sobre o carril metálico .....	65
56. Acomodação e localização dos dois pares de recetores dos sensores infravermelhos <i>BreakBeam</i> .....	66
57. Interface gráfica do servidor Web e código resultante na consola no lado do Cliente .....	67
58. Lista de APs disponíveis e código resultante se ligação ao AP escolhido tiver sido bem-sucedida .....	67
59. Troca de mensagens WebSocket entre cliente (esquerda) e servidor (direita) de inicialização .....	68
60. Pings entre Servidor e Cliente .....	68
61. Número de quebras de feixe (ou seja, pilares ultrapassados pelo protótipo) detetadas pelos dois pares de sensores, até a detecção do pilar de referência .....	69
62. Estado inicial e informação do número de quebras detetadas por cada sensor .....	69
63. Transições entre os diversos estados do programa .....	70
64. Estado 0 e estado 1 via CW .....	71
65. Estado 0 e estado 1 via CCW .....	71
66. Estado 2 e Estado 3 (à esquerda D1 e à direita D2) .....	71
67. Estado 4 – Sentido CCW .....	72
68. Estado 4 – Sentido CW .....	72

## LISTA DE TABELAS

1.	Excerto das características de algumas das testbeds que suportam mobilidade [28] .....	3
2.	Vantagens e desvantagens RFID conforme o intervalo de frequência utilizado [40] .....	15
3.	Principais características do sistema Tiny RTK GPS da <i>drotek Eletronics</i> .....	23
4.	Principais características do sistema TRC Beacon .....	26
5.	Precisão e custo associado ao sistema CONAC™ .....	28
6.	Resumo das características de um PIR da adafruit [22] .....	48
7.	Principais características de um conjunto <i>IRBreakBeam</i> [21] .....	49
8.	Principais características da board NodeMCU devKit .....	56



# 1 INTRODUÇÃO

A comunicação entre dois dispositivos e respetivo envio de informação data de há muitos anos, por volta dos anos 30, onde estes eram de dimensões tão elevadas que apenas recorrendo a meios de transporte terrestres ou marítimos era possível a sua deslocação. No final desta década surgem as primeiras soluções portáteis que permitem ser transportadas por uma única pessoa, o famoso “*walky-talky*”, um simples dispositivo de comunicação via ondas rádio. Com a chegada dos transístores e novos avanços na tecnologia aliada à eletrónica cada vez mais estes dispositivos rádio se tornam mais diminutos em dimensão de tal forma que num chip de um tamanho semelhante a uma moeda, se consegue englobar o *transciever* (dispositivo capaz de transmitir e receber informação) rádio e todo o circuito eletrónico necessário para o correto funcionamento do dispositivo (exemplo ilustrado em [23]). Surge, portanto, o conceito de **mobilidade** nos sistemas de comunicação. No entanto, como dito em [24]:

*“Mobilidade requer adaptabilidade...”*

A forma como os sistemas de comunicação atuais se auto configuram de modo a manter um registo do movimento e por sua vez a localização de um nó móvel e de importância fulcral para que a implementação do conceito mobilidade seja aplicável.

Nas redes de comunicação de hoje em dia, a rede divide o campo em que fornece cobertura de rede aos seus clientes em diferentes células controladas por uma estação base, tendo como objetivo principal cada uma destas estações base comunicar com os dispositivos móveis (nós) dentro dessa célula via ondas rádio. Neste tipo de redes celulares, o único problema associado à temática da mobilidade recai na situação em que o nó móvel se desloca de uma célula para outra, denominado por “*Cellular Handoff*”. As estações base das células em questão, através de protocolos robustos, permitem que o movimento entre células seja efetuado de uma forma contínua e sem que ocorram distúrbios na comunicação.

Num modelo alternativo, um outro tipo de redes existente é o conjunto denominado por MANET (Mobile Ad hoc Wireless Networks) [25], que consiste numa rede sem fios sem uma infraestrutura fixa capaz de se reconfigurar. Todos os nós constituintes deste tipo de redes atuam como routers e podem mover-se arbitrariamente. Neste tipo de redes a comunicação de informação é reencaminhada de router em router (o mesmo que dizer, de nó em nó) até ao destinatário final. É de fácil conclusão que caso os routers intermédios à fonte e ao destinatário da mensagem a ser transmitida se movam, a mobilidade irá afetar de uma forma significativa este tipo de redes. No entanto não existem apenas inconvenientes aliado ao facto de se introduzir mobilidade nos sistemas de comunicação. Em [26], descreve-se como a empregabilidade de mobilidade em redes MANET se encontra interligada com questões de segurança. Desta forma, seria também possível associar alguns aspetos positivos à introdução de mobilidade nestes tipos de sistemas.

## 1.1 MOTIVAÇÃO/ENQUADRAMENTO

Na secção anterior, referenciaram-se as redes MANET e inferiu-se que a ausência de uma infraestrutura fixa pode ser prejudicial à implementação de cenários de mobilidade. Portanto, a situação ideal seria a possibilidade de se desenvolver uma rede sem fios constituída por uma infraestrutura fixa, sendo esta composta por diversos nós tanto fixos como móveis, suportando assim mobilidade. Esta rede seria alvo dos mais diversos estudos ou para a prática das mais diversas experiências podendo assim avaliar-se o seu desempenho tanto em cenários reais como emulados. Estas redes existem atualmente e designam-se por *testbeds* (redes de testes) e o seu número tem crescido de forma bastante célere. Este crescimento deve-se principalmente ao elevado interesse por parte da comunidade científica responsável pela investigação e desenvolvimento de novas tecnologias a serem aplicadas nestes tipos de redes, sempre tendo em mente o foco principal – a avaliação dos mais distintos cenários reais e emulados. Existem inúmeras redes de testes atualmente e isto deve-se a um simples facto: cada rede de testes é única, ou seja, cada uma destas redes tem um propósito e uma infraestrutura característicos. Por exemplo em [27], descrevem-se e estudam-se vários exemplos de *testbeds*, entre elas a desenvolvida pela Universidade de Carnegie Mellon (CMU-DSR) constituída por cinco nós móveis e dois estáticos numa área abrangente de 700 por 300 metros. A mobilidade é implementada através de introdução de computadores portáteis transportados por veículos automóveis. Uma tabela que resume as características de um número diferenciado de testbeds é apresentada em [28], e replicada na Tabela 1.

Nome da Testbed	Arquitetura	Tipo de Modelação Móvel	Meio de modelação Sem Fios	Número de Nós Físicos
<b>APE</b>	Testbed do Mundo Real, (controlo distribuído)	Real (pessoa), conectividade lógica	IEEE 802.11	37
<b>Ritter et al.</b>	Testbed do Mundo Real	Real (pessoa)	Bluetooth + Módulo RF 433/868 MHz	5
<b>Maltz et al.</b>	Testbed do Mundo Real	Real (pessoa + veicular)	IEEE 802.11	8
<b>MiNT</b>	Testbed do Mundo Real, (miniaturizada)	Real (robôs)	IEEE 802.11	8

Tabela 1. Excerto das características de algumas das testbeds que suportam mobilidade [28].

Uma destas redes de testes encontra-se instalada no terraço do edifício Instituto de Telecomunicações (IT) em Aveiro [29], e é designada por **AMazING** (*Advanced Mobile wIreless Network playGround*) [30].

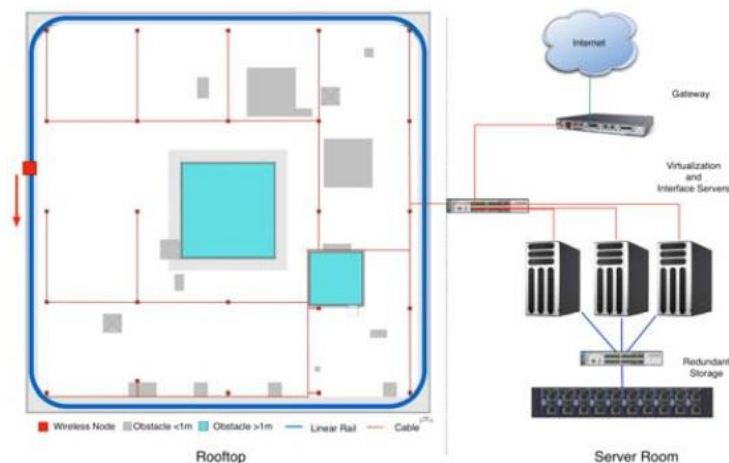


Figura 1. Esquemático da rede de testes AMazING.

Esta rede de testes surge após o conceito proposto pelo professor Rui Aguiar de uma rede de testes capaz de se distinguir de todas as outras devido à sua elevada diferenciabilidade de funções de controlo oferecidas ao utilizador e de um diverso leque de testes possíveis, mesmo de eventos de mobilidade. A plataforma, desenvolvida pelos membros do grupo ATNoG (*Advanced Telecommunications and Networks Group*) [31], consiste numa rede sem fios de livre acesso constituída por 24 nós fixos, como já dito anteriormente, no terraço do IT. Servidores de suporte e material redundante de armazenamento encontram-se no núcleo da testbed, pois estes servidores servirão como fonte de processamento para o escalonamento e manio das diversas experiências realizadas tal como a análise posterior dos resultados. Atualmente, um carril metálico foi instalado no terraço do IT, no qual está a ser aplicado um nó móvel capaz de atingir velocidades próximas de 60 km/h. Todo o esquemático da rede de testes AMazING e a interligação com o carril metálico e respetivo nó móvel, podem ser verificados na Figura 1.

## **1.2 OBJETIVOS DO TRABALHO**

Como é possível constatar na Figura 1, sobre um carril metálico (a cor azul) será colocado um nó móvel (a cor vermelha) capaz de o percorrer ao longo de todo o seu perímetro. O principal objetivo deste documento passa por uma proposta de um sistema de localização para que se conheça a posição do nó ao longo de todo o carril. De tal forma, os principais focos deste documento podem ser divididos nos seguintes:

- Estudo das diferentes técnicas ou estratégias de sistemas existentes nas mais diversas áreas de engenharia e telecomunicações capazes de localizar um objeto móvel;
- Escolha dos materiais adequados a uma estratégia viável para a infraestrutura montada no terraço do IT;
- Apresentação da solução proposta para o sistema de localização;
- Realização de testes e respetiva análise dos resultados utilizando a ideia proposta;

## **1.3 ESTRUTURA DO DOCUMENTO**

Este documento encontra-se dividido em capítulos, um Capítulo 2 onde se expõem e estudam as diversas estratégias existentes nas mais diversas áreas de engenharia, desde a navegação via satélite até aos sistemas de localização em robótica. No Capítulo 3 discutem-se os principais dispositivos utilizados nestes mesmos sistemas. No Capítulo 4 encontra-se descrita a solução implementada para a realização de um sistema de localização para se obter a posição do nó móvel que se desloca ao longo do carril metálico, desde os materiais utilizados até à implementação de todo o sistema. O Capítulo 5 apresenta todo o conjunto de testes realizados e resultados obtidos desses mesmos testes. Finalmente num capítulo 6 retratam-se algumas conclusões sobre os resultados obtidos e uma pequena análise crítica é realizada.





## 2 ESTADO DA ARTE

### 2.1 UMA BREVE INTRODUÇÃO HISTÓRICA.

O ano é 1500. O povo português encontra-se perante a sua, embora que acidental, maior descoberta marítima da sua longa história ao descobrir o Brasil na época também conhecida por Era dos Descobrimentos. Para que tal feito fosse possível, muito contribuíram técnicas de navegação marítimas que permitiram uma bem-sucedida, embora atribulada, viagem marítima até este país. De realçar que nesta era, os ditos navegadores, apenas com o conhecimento das constelações estelares e fazendo uso de instrumentos como o astrolábio (objeto que permitia obter a latitude de um navio no oceano, Figura 3) conseguiram feitos como o descrito anteriormente e posteriormente o descobrimento da Índia e outras zonas mais ocidentais [32].



Figura 2. Chegada ao Brasil por parte da armada Portuguesa [1].

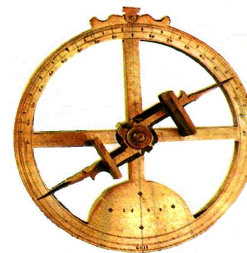


Figura 3. Astrolábio [71].

Contudo, sendo um ser criativo como é o ser humano, ao longo da sua história sempre se preocupou em inovar e melhorar as tecnologias já existentes relacionadas com a navegação nas mais diversas áreas. A criatividade aliada à curiosidade, também característica ao Homem, orientou-o para uma das maiores inovações tecnológicas da sua história, a exploração espacial. Este enorme salto tecnológico permitiu implementar uma diferente forma de localizar um objeto em qualquer parte do mundo, ou seja, uma técnica

que se possa denominar global. Recorrendo a satélites, colocados em órbita à volta da Terra em missões espaciais anteriormente realizadas foi implementado, um sistema de posicionamento global via satélite.

## 2.2 LOCALIZAÇÃO – O QUE É? A CHEGADA DE UM NOVO SISTEMA.

De acordo com [33], **localização** entende-se por ( “*lugar em que está localizado algo ou a ação e o efeito de localizar...*” ). Tendo em conta a afirmação anterior e com a recente descoberta tecnológica que permitiu colocar vários satélites em órbita da Terra, o **sistema de navegação global via satélite** é criado. Também denominado pela sigla **GNSS** (*Global Navigation Satellite System* em inglês) são vários os sistemas existentes atualmente que podem ser caracterizados como GNSS. Entre eles: o *GPS* (Estados Unidos da América), *Galileo* (União europeia), *GLONASS* (Rússia) e o *BeiDou* ou *COMPASS* (China). Neste presente momento apenas dois (GPS e GLONASS) se encontram totalmente operacionais estando, no entanto, para breve a total funcionalidade dos sistemas Galileo e COMPASS nos próximos anos.



Figura 4. Os diferentes sistemas GNSS e respetivos logotipos [2].

Um sistema GNSS é constituído por três grandes segmentos: um denominado por **espacial**, onde se encontram as constelações de satélites que formam o sistema e que são responsáveis pela transmissão dos sinais via rádio frequência, fundamentais para o cálculo da localização do objeto na Terra. Um segundo designado de **controlo**, pois como o nome indica, exerce funções de controlo sobre a informação recebida por parte dos satélites. Composto por várias estações de controlo e uma Master (estação com maior responsabilidade de todo o conjunto) proporciona uma correta informação aos satélites sobre questões temporais e de que órbita efetuar. De seguida recambia esta informação para uma unidade (*Master*) de controlo, responsável pela correção desta, a qual posteriormente é enviada de novo para os satélites no espaço.

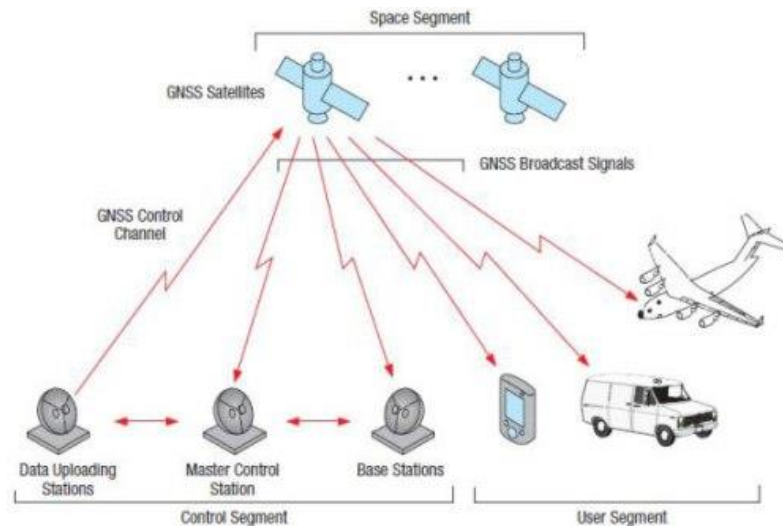


Figura 5. Segmentos de um sistema GNSS [3].

Finalmente, resta apenas o segmento reservado ao **utilizador** o qual compreende todos os utilizadores do sistema e respetivos recetores capazes de receber informação via rádio frequência. O número de utilizadores em simultâneo é ilimitado.

Um dos sistemas mais usualmente adotado para o prisma de localização de um objeto é um dos anteriormente referidos, designado por *Global Positioning System* (GPS – em português, Sistema de Posicionamento Global). Este sistema foi criado pela força área americana no ano de 1960, sendo adotado pelo exército americano alguns anos mais tarde. Como dito anteriormente no subcapítulo 2.2, com o lançamento de satélites para o espaço nos anos de 1978 o sistema é posto em total funcionamento no ano de 1995 [34].

Sendo um sistema composto por um conjunto de 24 satélites, colocados a cerca de 20 mil quilómetros de altitude, que circulam a Terra a cada 12 horas, permite obter uma estimativa da velocidade, posição global e até informações temporais de um determinado objeto na superfície terrestre. Cada um destes satélites colocados em órbita terrestre, transmitem sinais rádio com códigos únicos sendo desta forma facilmente identificados pelo recetor GPS. Através destes sinais codificados oriundos dos satélites, é possível de ser efetuada uma estimativa do tempo despendido entre a transmissão e a receção dos mesmos no recetor. Por sua vez, este intervalo de tempo (vezes a velocidade da luz) permite saber a que distância se encontra o satélite do recetor GPS.

Após se conhecerem as posições de pelo menos 4 satélites, o recetor utiliza a técnica denominada de **Trilateração** para obter uma estimativa da posição do objeto (esta técnica será discutida mais em detalha num ponto mais avançado deste documento).

No entanto, nem tudo é ótimo em relação ao sistema de posicionamento global. Estes tipos de sistemas padecem de possíveis **erros** aquando da receção dos sinais oriundos dos satélites por parte dos recetores colocados à superfície terrestre. Entre as principais causas de erros encontram-se os seguintes:

- Erros **oriundos do próprio utilizador** – até o corpo humano pode contribuir para erros de leitura por parte do recetor. Se um utilizador colocar o recetor a uma distância muito curta do seu corpo, pode bloquear parte dos sinais enviados pelos satélites diminuindo assim a precisão na estimativa da posição.
- Erros de **relógio dos satélites** – possíveis de ocorrerem devido a pequenas discrepâncias entre os relógios internos de cada satélite. Erros que são corrigidos na Estação Master de Controlo (ver Figura 5).
- Erros **orbitais** – as órbitas efetuadas pelos satélites encontram-se interligadas à altitude, posição e velocidade de cada satélite. As órbitas efetuadas pelos satélites podem, por exemplo, sofrer alterações nas suas órbitas devido a mudanças na força gravitacional com que estes são atraídos pelo planeta. Estes tipos de erros são monitorizados e corrigidos, novamente na estação de controlo.
- **Interferências atmosféricas** – estes tipos de interferências provocam refrações dos sinais rádio oriundos dos satélites GPS, enquanto atravessam as diferentes camadas de atmosfera da Terra. Estes sinais sofrem assim uma aceleração ou um retardamento, o que por fim resulta num cálculo impreciso da posição dos satélites por parte dos recetores GPS, que se encontram no terreno. Normalmente, a este tipo de erros está-lhes associado um erro não superior a 10 metros.
- **Interferência multicaminho (*Multipath*)** – ocorrem quando existem sinais rádio refletidos por parte de superfícies adjacentes ou próximas dos recetores GPS, que podem interferir ou chegarem mesmo a serem confundidos com o sinal verdadeiro, caso existisse uma transmissão sem reflexões. Um exemplo do fenómeno de *multipath* pode ser verificado na figura seguinte.

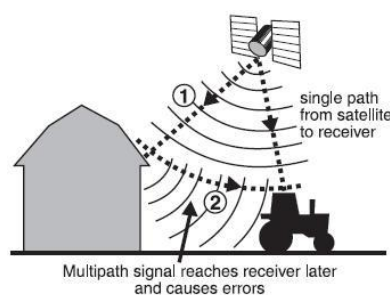


Figura 6. Erro devido ao fenómeno de multipath [4].

- **Número de satélites visíveis** – quanto maior o número de satélites que o recetor consegue “ver” melhor a precisão do sistema. A receção dos sinais pode encontrar-se bloqueada por terreno, edifícios, etc. Resumindo, quanto mais desimpedida ou clara a linha de transmissão entre satélite e recetor melhor para o funcionamento do sistema.

- **Geometria dos satélites** – refere-se à posição relativa dos satélites num determinado instante de tempo. Considera-se ideal quando os satélites se encontram afastados uns dos outros efetuando ângulos amplos entre eles. Uma fraca geometria existe quando os satélites se encontram numa linha ou se formam um aglomerado.

Concluindo, os sistemas de navegação via satélite (no qual se engloba o GPS) podem ser considerados os pioneiros na temática de localização e estimativa de posição de um objeto munido de movimento nas mais diversas áreas de aplicação.

### 2.3 UM OUTRO SISTEMA DE LOCALIZAÇÃO. A IDENTIFICAÇÃO RÁDIO FREQUÊNCIA

A identificação por rádio frequência, (*Radio Frequency Identification* – RFID - abreviado do inglês), como o próprio nome indica, faz uso de ondas rádio para a correta identificação e monitorização de objetos. Embora o sistema GPS também use ondas de rádio, normalmente o termo RFID só é aplicado a sistemas locais terrestres. Um sistema completo RFID tem três componentes essenciais: *RFID Reader*, *RFID Tags* e uma Antena (ou antenas conforme a aplicação assim o exija). A informação trocada entre as Tags e os Readers irá ser recambiada para um computador onde será processada e utilizada para os fins pretendidos pelo utilizador. Para que se tenha uma ideia mais clara que componentes são estes e como interagem entre si, segue-se um simples esquema ilustrativo de um sistema RFID na figura abaixo:

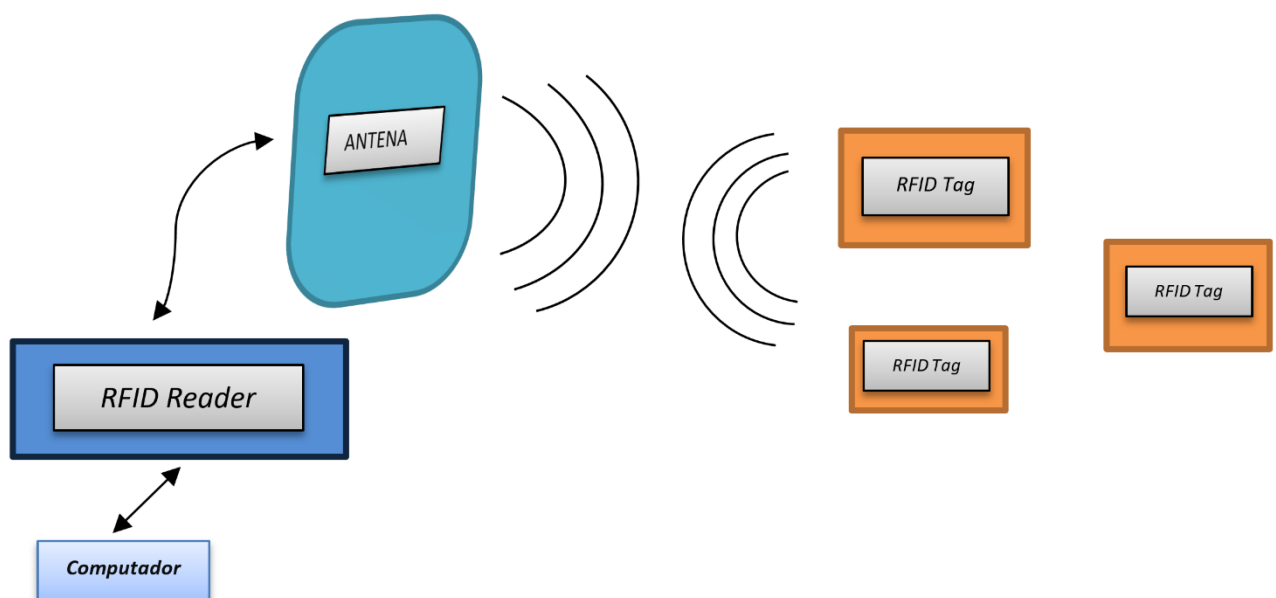


Figura 7. Um simples sistema RFID.

Em relação aos componentes constituintes de um sistema *RFID*, pode-se começar por descrever o “cérebro” deste tipo de sistemas, o ***RFID Reader*** (muitas vezes também designado por interrogador). Como exemplificado na figura anterior, é o elo de ligação entre as *Tags* e o utilizador que se encontra num computador onde a informação irá ser processada. Este dispositivo, através de antenas, permite que os diferentes objetos munidos de *Tags* ao se cruzarem numa vizinhança próxima dele o suficiente, sejam corretamente lidos/identificados. Este dispositivo é também capaz de escrever informação em memória das *Tags* que estejam prontas para esse efeito. Para este efeito existem três tipos de leitores possíveis de implementação: fixos, manuseáveis e integrados.

**Fixos** - como o próprio nome indica, encontram-se estacionários numa localização específica aquando da identificação e leitura das *Tags*. Entre as principais vantagens que oferecem encontram-se a possibilidade de escalabilidade no número de antenas, pois permitem um número reduzido tal como duas antenas até um máximo de trinta e duas. Portanto estes tipos de leitores podem ser opção em cenários em que a complexidade da área de cobertura e flexibilidade na escolha, no número e no tipo, diferente de antenas sejam fatores importantes. Em relação ao meio de comunicação preferido por este tipo de leitores para a transferência e envio de informação, a escolha prende-se entre dois tipos: caso se escolha via cabo, através do padrão ***Ethernet*** [35], ou então via sem fios através do padrão de rede 802.11, mais conhecido como ***WiFi*** [36]. Exemplares dos dois tipos encontram-se apresentados nas figuras seguintes.



Figura 8. Exemplar de um *Reader Ethernet* [5].



Figura 9. Exemplar de um *Reader WiFi* [6].

**Manuseáveis** – permitem ser transportados enquanto fazem a leitura dos variados objetos. Os leitores RFID manuseáveis são dispositivos que aglomeram o leitor e a antena embutida no mesmo. Entre os meios de comunicação mais usualmente utilizados entre leitor e *Tags*, incluem-se *Bluetooth* e *WiFi*. Um exemplo que faz jus da tecnologia *Bluetooth* como forma de via de comunicação para respetiva identificação das *Tags* encontra-se na figura abaixo.





Figura 10. Exemplo de um *RFID Reader Handheld* [7].

**Integrados** – dispositivos constituídos por um leitor e uma antena encapsulada e que eventualmente possibilitam a adição de uma outra antena, caso o seja necessário. Encontram aplicação em cenários onde o custo do leitor seja um fator importante, pois são de baixo custo e no entanto possibilitam uma implementação com uma ou duas antenas.

De seguida apresenta-se um exemplo de um leitor integrado, que utiliza como meio de comunicação para a transmissão e receção de dados tecnologia WiFi. Da figura seguinte, é de salientar as portas de Ethernet (a qual permite além de interface de comunicação, também pode ser utilizada como fonte de alimentação para o leitor através do padrão *Power over Ethernet (PoE)* [37]) e a porta adicional para uma antena, caso seja necessário.



Figura 11. *RFID Reader Integrado* [8].

Descreva-se agora outro componente dos sistemas RFID, o qual seja talvez o mais importante de todo o sistema RFID. **RFID Tags** (ou em português, Etiquetas) são usualmente constituídas por duas partes: um **chip** RFID (ou um CI - Circuito Integrado) no qual a Tag guarda informação em memória e um número único característico capaz de a identificar pelos leitores RFID que se encontrem nas redondezas. A segunda parte recai numa **antena** que consiga transmitir e receber toda esta informação, a qual costuma estar



embutida numa superfície que engloba tanto a antena como o chip. Uma outra forma de caracterizar sistemas RFID recai no facto de as suas *Tags* possuírem ou não fontes de energia. Sistemas que possuam uma fonte interna de energia capaz de alimentar a Tag aquando da resposta de um sinal gerado pelo Reader, denominam-se **Ativos**. De outro modo, sistemas que não possuam qualquer fonte de energia interna e apenas respondam a sinais oriundos dos leitores devido à energia transmitida pelos mesmos, designam-se de **Passivos**.

Caso a antena recolha também a energia necessária para ligar o chip da Tag, o sistema classifica-se como sendo ativo. Tipicamente quanto maior for a área por ocupada pela Tag maiores serão as quantidades de energia apreendidas e recambiadas para o chip, o que se exprime num aumento na capacidade da deteção por parte da Tag. De acordo com o tipo de superfície onde serão aplicadas, seja ela metálica ou que possua elevados índices de humidade, existem *Tags* próprias com o intuito de superar essas adversidades.

De seguida apresentam-se dois exemplares de *Tags* que podem ser utilizadas nestes cenários:



Figura 12. *Metal Tag* [9].



Figura 13. *Wet Tag* [10].

Em relação ao último, mas não menos importante, elemento que falta retratar de um sistema RFID, as **antenas** são dispositivos que convertem corrente elétrica em ondas eletromagnéticas. Estas mesmas ondas são depois irradiadas no espaço onde serão depois captadas pela antena do outro conjunto (leitor ou Tag) que as converte novamente para energia elétrica. Existem dois tipos de antenas, diferenciando-se de acordo com a sua polarização, **lineares** ou **circulares**.

Antenas lineares emitem radiação, via ondas rádio, ao longo de um simples plano no espaço (Figura 15), permitindo uma longa distância entre *Reader* e *Tag* e elevados níveis de energia. Este último aspeto permite que os sinais por si enviados, penetrem em materiais mais rígidos possibilitando assim a leitura das *Tags* encapsuladas nesses materiais. Este tipo de antenas é sensível à orientação das *Tags*, ou seja, a *Tag* necessita de estar enquadrada com o feixe irradiado pela antena. Dependendo do posicionamento e do ângulo da *Tag* possíveis erros na leitura podem surgir, o que resulta num inconveniente deste tipo de antenas.

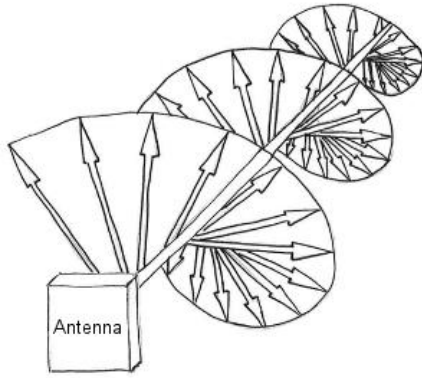


Figura 15. Polarização circular [72].

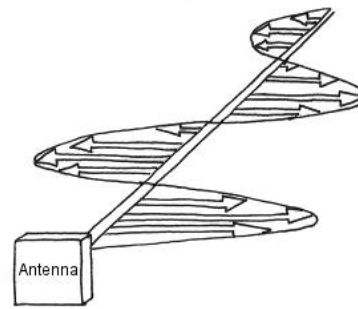


Figura 14. Polarização linear [72].

Pelo contrário, antenas circulares dividem a energia em dois planos (Figura 14). Devido ao facto de a energia ser dividida possuem uma deteção, em termos da distância a que o Reader se pode encontrar da Tag, inferior às antenas lineares.

Os sistemas RFID operam normalmente entre três grandes bandas de frequência. Em cada uma destas bandas de frequência as ondas rádio comportam-se de maneiras distintas, pelo que cada banda possui as suas vantagens e desvantagens na altura de ser escolhida. Entre essas bandas ou *intervalos de frequência* disponíveis para os sistemas RFID encontram-se:

- *LF (Low Frequency)* – intervalo para as baixas frequências, entre (125-134kHz);
- *HF (High Frequency)* - intervalo para as altas frequências, entre os (3-30) MHz. Embora o valor mais usual de operação seja de 13,56 MHz neste intervalo;
- *UHF (Ultra High Frequency)* – intervalo para as ultras altas frequências, disponíveis numa gama de 433 MHz e (856-960) MHz. Existem ainda dois subconjuntos, para o intervalo (856-960), de acordo com a localização geográfica em que se encontra o sistema RFID a operar:
  - O padrão FCC (*Federal Communications Comission*)[38] – utilizado por toda a América do Norte, tal como a maioria dos países da zona das Caraíbas e grande parte da América do Sul, num intervalo de frequências compreendido entre os (902-928) MHz;
  - O padrão ETSI (*European Telecommunications Standards Institute*)[39] – utilizado por toda a União Europeia e países aderentes das normas europeias, num intervalo de frequências compreendido entre os (865-868) MHz.

Sistemas que operem a frequências mais baixas possuem um menor alcance e uma taxa de atualização de leitura de informação mais lenta. No entanto são capazes de ler informação de *Tags* colocadas em ou próximas de superfícies metálicas ou em ambientes com elevados fatores de humidade. Por outro lado, sistemas RFID que optem por frequências mais elevadas, como expectável, possuirão uma transferência de dados mais célere e permitem a leitura de objetos a distâncias superiores quando comparados com sistemas que utilizem frequências mais baixas. Contudo são mais vulneráveis a interferências provocadas por metais e líquidos existentes no meio ambiente.

Na tabela seguinte, encontram-se resumidos alguns dos aspetos-chave de cada intervalo de frequência:

Intervalo de frequência (usadas em RFID)	LF (Low Frequency) (125-134kHz)	HF (High Frequency) (13,56MHz)	UHF (Ultra High Frequency)	
			Sistemas Ativos (433 & 856-960)MHz	Sistemas Passivos (856-960)MHz
<b>Aplicações</b>	Identificação Animal; Ambientes com elevados índices metálicos ou líquidos.	Cartões de identificação; Métodos de pagamento	Monitorização de objetos de larga escala; Construção civil; Sistemas de navegação aérea;	Monitorização de bens alimentares ao longo de uma cadeia de produção; Autenticação de produtos farmacêuticos.
<b>Alcance</b> (Máximo de leitura entre Reader e Tag)	Em contacto com o objeto ou até um máximo de 10 centímetros	Entre 10cm a 1m.	Entre 30 a 100 metros ou mais	Em contacto com o objeto ou até 25 metros ou mais
<b>Custo</b> De acordo com o alcance	0,5 até 5 dólares	0,23 até 10 dólares	25 até 100 dólares	0,13 até 25 ou mais dólares

Tabela 2. Vantagens e desvantagens RFID conforme o intervalo de frequência utilizado [40].

Constata-se através da Tabela 2, que dependente da aplicação, alcance e custos máximos pretendidos a escolha do intervalo de frequência a ser utilizado para o sistema RFID requer um compromisso entre estes aspetos fundamentais para que se obtenha um resultado otimizado.

Concluindo, um breve aglomerado das principais vantagens e desvantagens deste tipo de sistemas é apresentado de seguida. Em termos de vantagens:

- A identificação das *Tags* não requerer intervenção humana, resultando assim numa redução nos custos de implementação e elimina prováveis erros humanos originados aquando da aquisição de dados;

- Este tipo de sistemas não requer uma linha de visão desobstruída entre Reader e Tag(s), ou seja, menores restrições aquando na sua implementação;
- O alcance a que se consegue uma leitura admissível com este tipo de sistemas, dependendo do tipo de componentes utilizados, é um fator bastante favorável;
- As *Tags* possuem capacidades de leitura/escrita em memória, o que lhes permite armazenar grandes quantidades de informação. Para além que ao possuírem um identificador único, permite a estes sistemas que os objetos sejam identificados individualmente;
- As *Tags* são resistentes a condições atmosféricas adversas, como por exemplo, pó, metais, líquidos, etc;
- Múltiplas *Tags* podem ser lidas simultaneamente, o que aumenta substancialmente a eficácia deste tipo de sistemas;

Em relação às desvantagens deste tipo de sistemas, temos:

- Colisões entre sinais ao efetuarem-se leituras simultâneas podem surgir e para que tal seja evitado algoritmos anti colisão são implementados resultando em custos adicionais;
- O custo associado, como é possível analisar pela Tabela 2, está dependente do tipo de *Tags* utilizado. Para aplicações que requeiram uma monitorização de objetos a distâncias elevadas, o custo total do sistema pode tornar-se elevado;
- Possíveis pontos mortos podem surgir, ou seja, o sinal RF pode não chegar com força suficiente para que a comunicação entre *Reader* e *Tag* seja efetuada de forma correta. A solução para eliminar estes “pontos mortos” pode passar pela utilização de mais que um *Reader* ou antenas com polarização circular, resultando novamente em custos adicionais ao sistema base;
- Dependendo do tipo de aplicação do sistema, questões relativas a segurança e privacidade podem implicar que uma encriptação dos dados transmitidos seja efetuada antes de se iniciar a transmissão. Deste modo pessoas não autorizadas à utilização do sistema são impedidas de ler/escrever a informação guardada nas *Tags*.

## 2.4 LOCALIZAÇÃO EM ROBÓTICA

Atualmente, é de expectativa comum que a tecnologia evolui a um ritmo acelerado e que os novos sistemas de posição/localização desenvolvidos acompanharam também esta evolução. Um dos ramos a surtir particular efeito deste rápido crescimento tecnológico foi a **robótica** - ramo de engenharia que envolve todo o processamento desde a concepção, desenvolvimento e operação de robôs. O emprego destas máquinas inteligentes nas mais diversas áreas industriais (como por exemplo: agricultura, militar, hospitalar, automóvel, mineira, transportes, etc.) é notório nos dias que correm. Com o intuito de tornar estas máquinas autônomas e o mais verosímil ao ser humano, inúmeras estratégias de localização e orientação são desenvolvidas.

No ramo da robótica, robôs móveis são usualmente distribuídos em duas grandes categorias de acordo com o seu tipo de movimento. Uma **local** ou **relativa** e uma outra **global** ou **absoluta**. Ambas serão descritas de seguida, recorrendo também a alguns exemplos para cada uma delas.

### 2.4.1 LOCALIZAÇÃO RELATIVA / LOCAL

Na literatura atual, uma outra designação para localização relativa tem o nome de *deduced reckoning* ou **dead-reckoning** (em português cálculo deduzido, termo oriundo dos tempos da navegação marítima). Este processo consiste em determinar a posição de um robô móvel através de parâmetros como velocidade, tempo e direção, a partir de uma posição inicialmente conhecida. Estes parâmetros são capturados por um leque de sensores (dispositivos que serão mais tarde descritos) montados no robô.

Portanto, conhecendo-se previamente a posição em que se iniciou o movimento juntamente com os valores obtidos pelos sensores uma estimativa da posição é possível de ser obtida através de manipulação matemática desses mesmos valores (integrando-se a velocidade obtém-se posição).

Duas subcategorias quando combinadas formam a técnica denominada de *dead-reckoning*; entre elas encontram-se a **Odometria** e a **Navegação Inercial**.

**Odometria** usualmente encontra-se aliada a robôs que possuam movimento rotacional pois a estimativa da posição é calculada através da contagem das rotações que as suas rodas efetuam durante o seu movimento através de dados recolhidos por sensores colocados nos eixos de rotação. Considere-se a seguinte analogia para a técnica de Odometria: uma pessoa tendo conhecimento do seu ponto de partida, conta o número de passos que efetuou durante o seu trajeto ou durante o tempo que se movimentou. Essa pessoa, por exemplo, deu 20 passos, logo sabe que se encontra a 20 passos de distância do ponto de partida. Colocando agora o

problema no prisma de um robô, este contará o número de rotações que efetuou durante o intervalo de tempo que se movimentou. Sabendo que cumpriu, por exemplo, 200 rotações num determinado intervalo de tempo, calcula a sua velocidade e mais tarde a posição em que se encontra é obtida através da integração do valor da sua velocidade. Os sensores usualmente utilizados para a determinação da posição neste tipo de estratégia são designados por codificadores (*encoders* na terminologia anglo-saxónica). O seu modo de funcionamento e diferentes tipos serão estudados mais tarde neste documento (Capítulo 3).

Esta técnica padece de um grave problema, ao integrar-se a velocidade das rodas detetada pelos sensores de uma forma incremental também o ruído e outros possíveis erros associados (como por exemplo o deslize de uma das rodas do robô) irão ser acumulados ao longo de tempo resultando assim num aumento significativo do erro no posterior cálculo da estimativa da posição e orientação do robô.

Os **erros** oriundos desta técnica, denominada por odometria, podem ser categorizados em duas grandes categorias: erros **sistemáticos** e erros **não-sistemáticos** [19, p. 130].

Por erros sistemáticos entende-se tudo o que é característico ao robô, as suas propriedades cinemáticas como imperfeições no diâmetro de uma das rodas do robô ou um possível desalinhamento das rodas, por exemplo. Erros não-sistemáticos estão interligados a fenómenos que envolvam a interação robô-superfície, como por exemplo o deslize de uma das rodas ou a possibilidade de o robô ultrapassar lombas durante o seu percurso. Contudo, continua a ser uma das técnicas mais utilizadas, na área da robótica, devido a permitir elevadas taxas de amostragem e ser uma solução de baixo custo.

**Navegação Inercial**, como dito em [41], é uma técnica inicialmente desenvolvida para auxiliar a localização na navegação aeronáutica, rapidamente foi adaptada para uso em missões aeroespaciais e mais tarde em aplicações de teor de transporte marítimo. Acelerómetros e giroscópios (ver secção 3.3) são as ferramentas de eleição para efetuar as medidas de aceleração e rotação, respetivamente. As medidas obtidas serão por sua vez integradas uma vez (ou duas vezes para acelerómetros) obtendo-se assim a posição.

Sistemas que recorram a esta solução disfrutam da vantagem de serem independentes de fontes externas, ou seja, dispensam qualquer referência externa para o seu correto funcionamento. No entanto como o número de erros irá ser também integrado ao longo do tempo, a localização relativa passa por não ser muito fiável para sistemas onde uma elevada precisão acerca da posição seja um requisito muito importante.

Concluindo, um exemplo de uma plataforma que integra estas duas técnicas de Odometria juntamente com o auxílio de ferramentas que permitam navegação inercial é descrita em [42]. Fazendo uso de três giroscópios, um acelerómetro tri-axial e dois sensores de inclinação (*tilt sensors* em inglês), esta plataforma consegue obter a posição de um robô móvel apenas com um desvio na ordem de 1 a 8 cm por segundo, dependendo da frequência com que existam diferenças nos valores da aceleração do robô.

## 2.4.2 LOCALIZAÇÃO GLOBAL / ABSOLUTA

Entende-se por localização absoluta ou global, o cenário em que a posição de um robô é calculada através de parâmetros exteriores ao robô, ou seja, está dependente de fontes externas. A posição é fornecida através de dispositivos, como por exemplo os “*beacons*” (faróis em português), marcos geográficos (naturais ou artificiais) ou sinais rádio via satélite (por exemplo o GPS). Recorrendo a diferentes sensores, estratégias distintas surgem. Entre as principais encontram-se as seguintes: “faróis ativos”, o *GPS*, navegação por marcos e finalmente a navegação baseada em mapas (“*Map-based Mapping*” ou “*Map-Matching*” em inglês).

### 2.4.2.1 FARÓIS ATIVOS

São sistemas que encontram a sua aplicação principalmente na área naval e navegação aérea, permitindo a determinação da posição de um robô móvel de forma fiável e com elevada precisão. Contudo, podem também ser aplicados na localização e determinação de uma estimativa para a posição de um robô móvel terrestre. Sistemas que se baseiem em faróis ativos são capazes de demonstrar uma elevada precisão e garantia na determinação da posição do robô utilizando uma capacidade de processamento mínima.

No entanto não existem apenas vantagens nestes tipos de sistemas, existem também desvantagens como por exemplo, um elevado custo na instalação e manutenção destes mesmos sistemas.

Os sistemas de faróis ativos dividem-se em dois diferentes tipos: a Trilateração e Triangulação.

A Trilateração é a determinação da posição de um robô móvel baseada na sucessiva medição de distâncias entre os faróis conhecidos e o robô. Estes sistemas recorrem a um conjunto de três ou mais faróis que funcionam como transmissores, montados em posições normalmente pré-estabelecidas e fixas, e um recetor no robô móvel para o qual se quer descobrir a posição. O exemplo recíproco também é possível de acontecer, em que o transmissor se encontra no robô e os recetores situam-se em sítios estratégicos na área abrangente. Utilizando a informação dos tempos-de-voo (*time-of-flight-TOF em inglês*) o sistema consegue computar a distância entre os transmissores estacionários e o recetor a bordo do robô. Um dos exemplos, talvez mais usual, destes sistemas, é o *GPS*, existindo outros como por exemplo os baseados em sensores ultrassónicos.

### 2.4.2.2 TECNOLOGIA GPS E O RTK

**Global Positioning System (GPS)**, como já descrito em algum detalhe na secção 2.2, é um sistema de posição via satélite que padece de alguns erros. Com isto em mente, uma abordagem diferente do sistema foi explorada. Essa técnica foi denominada de GPS Diferencial em Tempo Real (ou *Real Time Differential GPS – DGPS* na nomenclatura anglo-saxónica). **DGPS** baseia-se na premissa de colocar um segundo recetor GPS (denominado caracteristicamente por “*rover*”) numa vizinhança próxima do primeiro recetor (ao qual usualmente se dá o nome de “*base station*” ou estação base) normalmente dentro de um raio de 10 quilómetros. No entanto, segundo [41, p. 240], mesmo recorrendo a esta técnica no melhor dos casos a precisão na estimativa da posição encontra-se tipicamente à volta de 4 a 6 metros.

Como é de carácter da tecnologia, esta não parou de evoluir e deu aso ao surgimento de uma nova técnica de GNSS denominada *Real Time Kinematics – RTK* (Cinemática em Tempo Real). Esta tecnologia permite atenuar, se não mesmo remover, alguns erros que afetam as anteriores técnicas como GPS e DGPS, conseguindo obter uma precisão ao nível do centímetro. Tal como o DGPS, RTK utiliza novamente uma base station (que serve de referência encontrando-se fixa) e um rover recetor móvel. A base station efetua correções em tempo real para o rover através de um elo de ligação previamente estabelecido entre os dois. Estes dois elementos devem encontrar-se afastados entre si até um máximo de 20 quilómetros.

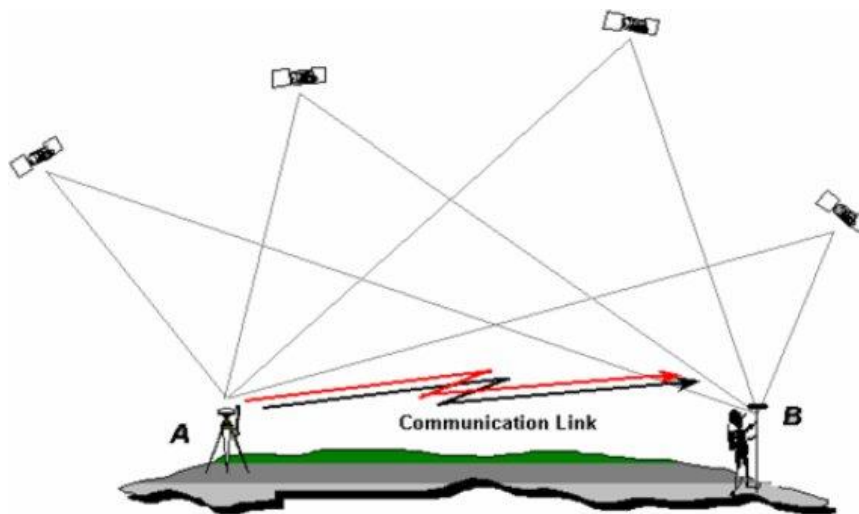


Figura 16. Diagrama de um sistema RTK [45].

Na figura anterior, podem-se identificar os dois elementos fundamentais deste sistema: na posição A (a *base station*) e numa posição mais afastada em B (o *rover* móvel).

A tecnologia RTK determina a posição do *rover* relativamente à estação de referência através da técnica que se designa por “*Carrier-Phase Positioning*” (Posicionamento via fase da portadora). A estação base calcula a fase da onda portadora em vez dos usuais códigos atribuídos a cada satélite (códigos *Pseudo-Random Number – PRN*, mais pode ser visto em sobre estes códigos [15, Appendix E].



Segundo [34], a utilização desta técnica traduz-se tanto em vantagens como em problemas: as vantagens prendem-se com o facto de tanto erros no recetor e provenientes da atmosfera serem minimizados. O problema principal recai sobre a necessidade de se resolver problemas de ambiguidade da fase portadora através de tentativa-erro.

É de fácil compreensão que todo este processamento em tempo real requer um esforço computacional bastante elevado e que na altura em que primeiro foi adotado (nos anos 80), a tecnologia existente nessa época poderia levar horas a completar todos os cálculos necessários. Ora, hoje em dia, com os avanços da tecnologia é possível efetuar tudo isto em tempo real, daí o nome Cinemática em Tempo Real (RTK). É neste âmbito que surge o sistema desenvolvido pela empresa *Drotek Eletronics*<sup>TM</sup>, que faz uso de tecnologia RTK (*Real Time Kinetics*) aliada ao módulo *NEO-M8P* da *u-blox*<sup>TM</sup> para obter uma precisão na estimativa da posição na ordem do centímetro.

Este módulo de alta precisão GNSS tanto pode albergar as funcionalidades de estação base e de rover (NEO-M8P-2) ou apenas de rover (NEO-M8P-0) [11].

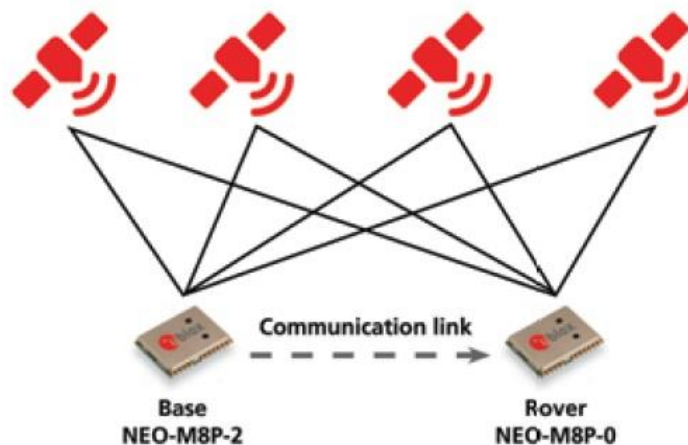


Figura 17. Ilustração de um sistema RTK utilizando os módulos NEO-M8P [11].

Na figura anterior é possível verificar os dois tipos de módulos fornecidos pela *u-blox*<sup>TM</sup> discutidos anteriormente num cenário de RTK. O módulo base envia sinais de correção através do protocolo RTCM (*Radio Technical Commission for Maritime Services* – entidade responsável pela transmissão de mensagens GNSS [(RTCM)]) para o módulo rover através de um elo de ligação (como por exemplo via radio frequência na gama *UHF*, *WiFi*, *Bluetooth*, *XBee*, etc.) o que permite que este esboce a sua posição relativamente à base até uma precisão ao nível do centímetro.

A *Drotek Eletronics*<sup>TM</sup> fornece diferentes formatos do seu sistema RTK de acordo com o tamanho, valores de alimentação, e interfaces de comunicação pretendidos. Mas entre eles, a opção que melhor se aplica à escolha de um sistema de posição é a versão denominada de *Tiny RTK*, a opção de menor tamanho e assim sendo a de menor custo associado. Como se pode verificar na figura seguinte, esta versão é bastante simples:

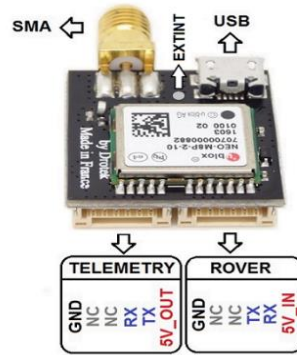


Figura 18. Board *Tiny RTK* da *drotek eletronic* (*Base station*) [45].

Na figura 18, é possível visualizar as entradas e saídas da *board* (placa impressa) *Tiny RTK*, tal como uma porta USB (o que lhe permite a funcionalidade de *plug and play*), uma saída SMA - (*SubMiniature version A* [44]) conector para uma possível antena e por último uma saída de Telemetria. Telemetria é o processo de transmissão de dados via um meio de comunicação sem fios entre duas entidades. Nesta situação a telemetria entre base e *rover* compreende-se na transmissão de dados via sem fios através de um destes meios de comunicação: *Bluetooth*, *WiFi*, *UHF*, *XBee*, etc.

É também passível de observação a incorporação do módulo de alta precisão GNSS NEO-M8P (neste caso específico o NEO-M8P-2, pois trata-se de uma *base*), o que permite funcionalidade de RTK à *board*.

Para que o sistema RTK seja formado, apenas são necessárias duas *boards* *Tiny RTK*. Uma servindo como *base station* – com o módulo NEO-M8P-2 – e outra funcionando como *rover* – com o módulo NEO-M8P-0. Uma ilustração do descrito anteriormente pode ser visualizado na figura seguinte:

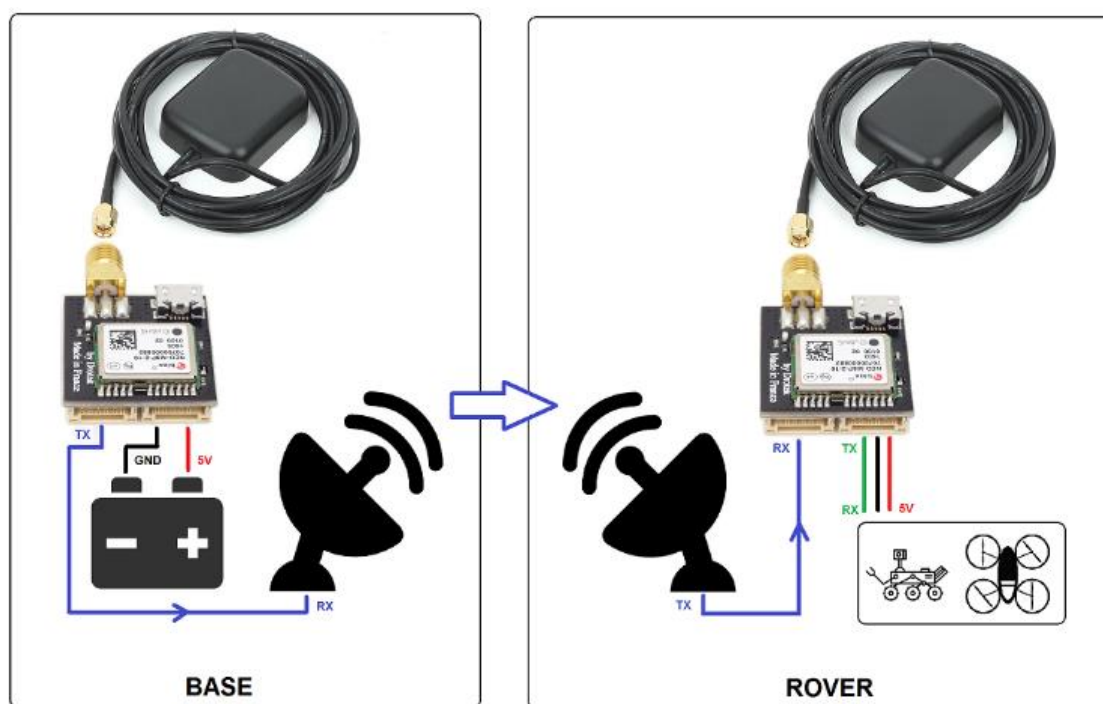


Figura 19. Sistema *Tiny RTK* da *drotek Eletronics* [45].

Na figura anterior, representa-se o funcionamento de um sistema *Tiny RTK*. A fonte de alimentação de 5 Volts tem de ser capaz de alimentar o módulo RTK, a antena GPS e a telemetria necessária para a comunicação entre base e *rover*. É através de telemetria que a informação é recebida pelo *rover*, o qual de seguida recalcula a sua posição centimétrica que posteriormente será lida pelo robô ou drone.

Resumindo, as principais características deste sistema encontram-se na tabela seguinte:

	Valor		Unidade
Precisão	2,5		cm
Características elétricas	5		V
	25		mA
Funcionalidades	Meio de comunicação	Todos os que possam ser interligados a uma UART: Bluetooth, XBee, WiFi, UHF, etc.	----
	Tamanho	23,5 x 24,5	mm
	Peso	5	g
	Custo total do sistema	Base (199.00) + Rover (189) = 388	€

Tabela 3. Principais características do sistema *Tiny RTK GPS* da *drotek Eletronics*.

Concluindo, este sistema oferece uma grande vantagem ao oferecer precisão na ordem dos centímetros, enquanto que a principal desvantagem pode passar pelo seu custo (médio-elevado), embora em alguns

cenários o custo do sistema pode ser justificado pela enorme vantagem de este mesmo possuir precisão centimétrica ao nível da posição.

### 2.4.2.3 SISTEMAS ULTRASSÓNICOS

Sistemas ultrassônicos trilaterais são sistemas capazes de oferecerem uma precisão média-alta na estimativa da posição do robô móvel, sendo também uma solução de baixo custo. Existem dois tipos de implementações: primeiro uma em que o robô transdutor (ver secção 3) transmite dele próprio para múltiplos recetores fixos, em posições previamente estabelecidas. E um segundo onde um único recetor se encontra à “escuta” no próprio robô com múltiplos transmissores fixos atuando como faróis. A primeira destas implementações encontra-se mais adequada para sistemas em que o número de robôs seja diminuto, enquanto que a segunda se aplica com maior eficácia em cenários em que o número de recetores fixos passivos seja indiferente, um pouco como o que se passa no GPS.

É na primeira implementação que se enquadra o *IS Robotics 2-D Location System* da companhia *IS Robotics, Inc. (ISR)* [46, p. 427], um sistema ultrassónico de baixo custo, que permite aos seus robôs uma precisão na localização da posição de 1,27 cm num quadrado de 9 x 9 m de área em que o robô opere. Recorrendo a uma “master” (mestra) base station ligada através de cablagem a dois “pingers” ultrassónicos que atuam como “slaves” (escravos), compreendidos entre uma distância de cerca de 2 metros. A comunicação entre os dois elementos fundamentais deste sistema (Base Station e robôs) é conseguida através de uma ligação via RF na gama *UHF* (902-928 MHz). Uma ilustração do sistema é apresentada de seguida:

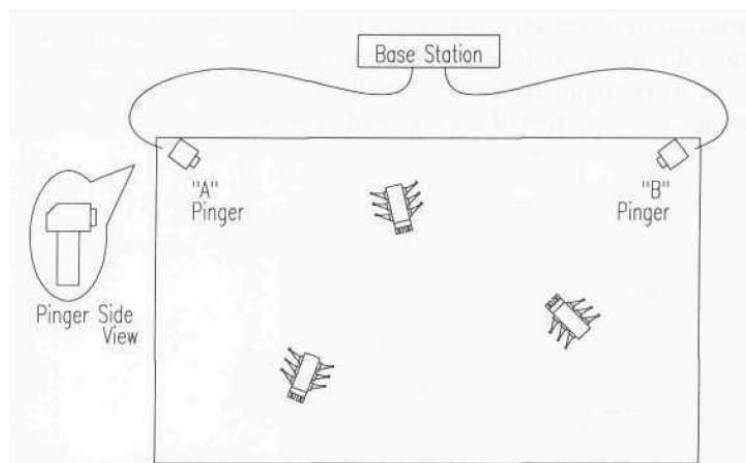


Figura 20. Sistema Ultrassónico da *IS Robotics* [19].

O tempo despendido entre a receção e a deteção dos pulsos ultrassónicos é utilizada para calcular a posição dos robôs na área de operação.

Triangulação, nesta configuração existem três ou mais transmissores ativos localizados em posições conhecidas. O robô móvel equipado com um sensor rotacional, regista os ângulos nos quais “vê” estes faróis transmissores relativamente ao seu eixo longitudinal. Desta forma, é possível calcular as suas coordenadas no plano  $xOy$ , juntamente com a orientação do robô que é inicialmente desconhecida. Obviamente também o caso recíproco em que o robô apenas se responsabiliza pela transmissão de informação, ou seja, ele próprio comportar-se como um farol deixando todo o processo de recolha e tratamento de informação para os recetores colocados em localizações estratégicas na área abrangente, é também possível. Um problema resultante desta implementação poderá recair sobre a necessidade se ter uma visão clara de pelo menos três faróis possibilitando a triangulação.

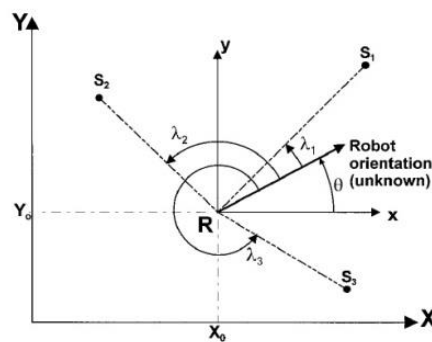


Figura 21: Medição dos três ângulos necessários para se obter triangulação [19].

De entre todos os sistemas disponíveis, os mais utilizados na temática da localização de robôs móveis são os que fazem uso da ótica para obter a triangulação. Estes sistemas caracterizam-se essencialmente por duas componentes: uma relacionada com os mecanismos de leitura/varrimento e outra com o conjunto de recetores fixos localizados estrategicamente no meio que os rodeia. De entre eles destacam-se os seguintes:

- Detetores de varrimento com faróis ativos servindo de emissores;
- Emissores/Detetores de varrimento com alvos retro refletivos passivos;
- Emissores/Detetores de varrimento com alvos ativos servindo de transponders;
- Emissores rotacionais com alvos fixos colocados em locais estrategicamente planeados;

É nesta última categoria que se enquadra o sistema desenvolvido pela: *Transitions Research Corporation (TRC)*, denominado por *TRC Beacon Navigation System* [19].

Este sistema consegue calcular a posição e direção de um robô móvel até distâncias de 24,4m desde que este esteja compreendido dentro de uma área quadrilátera (24,4 por 24,4 metros) definida por quatro faróis refletivos. Inicialmente, 15 segundos de visão desobstruída entre os quatro faróis e o robô são necessários para adquirir corretamente a posição inicial do robô, e após isso apenas dois faróis devem permanecer à vista do robô. De seguida o robô munido de um microprocessador dedicado fornece de forma contínua parâmetros de navegação ( $x$ ,  $y$ ,  $\theta$ ) ao controlador a bordo do robô via uma porta série do tipo RS-232. Uma unidade de varrimento colocada no robô atualiza até a uma taxa máxima de 1Hz com uma velocidade de

varrimento de cerca de 60 rpm (rotações por minuto), os sinais refletidos pelos faróis fixos aquando do movimento do robô. Uma ilustração do sistema pode ser observada na figura seguinte:

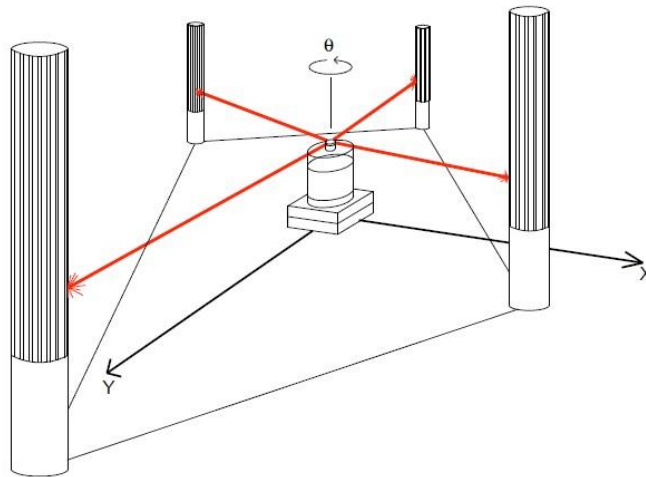


Figura 22: Demonstração de funcionamento do sistema de navegação *TRC Beacon Navigation System* [19].

De seguida, são apresentadas as principais características deste sistema, na seguinte tabela:

	Valor	Unidade
Precisão	120	mm
Requisitos de Alimentação	0,5 @ 12	[A] @ [V]
	0,1 @ 5	[A] @ [V]
Custo total do sistema	11,000	\$

Tabela 4. Principais características do sistema TRC Beacon.

Resumindo, uma linha de visão desobstruída entre transmissor e recetor deve ser mantida para uma correta localização da posição e orientação do robô para a correta utilização deste sistema.

Um tipo de sistema semelhante ao descrito anteriormente, o *MTI Research CONAC (Computerized Opto-electronic Navigation and Control)* <sup>TM</sup> [19], que se baseia numa rede de faróis ativos previamente instalados em posições fixas e pré-definidas foi desenvolvido pela entidade *MTI Research, Inc., Chelmsford, MA*. É um sistema de navegação de referência de baixo custo e elevado desempenho que utiliza uma unidade veicular equipada com um laser de varrimento denominado por *STRUCTURED Opto-electronic Acquisition Beacon (STROAB)*. O feixe emitido por este laser, é por sua vez difundido verticalmente de modo a permitir a montagem a alturas arbitrárias da rede de recetores designados por *Networked Opto-electronic Acquisition Datums (NOAD)*.

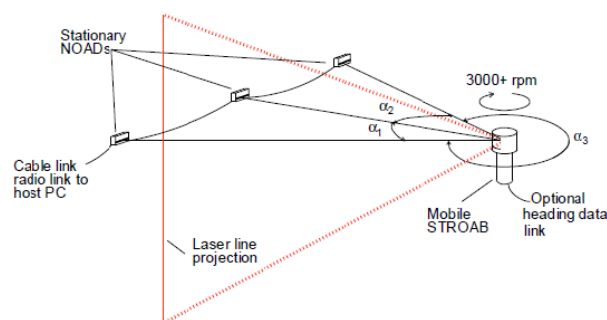
Figura 24. Exemplo de um *STROAB* [19].Figura 23. Exemplar de um *NOAD* [19].

A detecção de iluminação incidente nos *NOAD* reproduz-se numa resposta através da rede, constituída por este aglomerado de recetores *NOAD*, que será comunicada por sua vez a um computador anfitrião responsável pelo cálculo dos ângulos de incidência necessários para uma correta determinação da posição do robô. Estes *NOADs* que servem de referência para todo o sistema, usualmente, encontram-se interligados via cabo. No entanto, é possível alterar a comunicação entre *NOADs* via ondas rádio de modo a eliminar possíveis inconvenientes onde cablagem necessária do sistema não seja o cenário mais adequado, desde que a energia seja entregue de forma independente aos vários locais onde se situam os *NOADs*.

Este sistema com apenas um *STROAB* e três *NOADs* consegue cobrir de forma eficaz uma área total de 33,000 metros quadrados, desde que não haja estruturas a bloquear a “vista” entre os dois tipos de componentes. Este conjunto ideal de três *NOADs* do sistema é selecionado de forma dinâmica pelo computador anfitrião, baseando-se para tal na localização atual do robô e qualquer barreira visual predefinida.

De modo a minimizarem-se problemas de obstruções no campo de visão entre *STROAB* e *NOAD*, o peso-pluma *STROAB* (usualmente mais leve que 250 gramas) é normalmente montado tão alto como possível de preferência em mastros auxiliares.

A figura seguinte ilustra o modo de funcionamento deste sistema, onde se verifica a interação entre os dois principais componentes (*STROAB* e *NOAD*):

Figura 25. Sistema *CONAC™* [19].

Verifica-se a velocidade de rotação do *STROAB* (ordem das 3000 rotações por minuto) e a forma como o sistema deteta os ângulos de incidência ( $\alpha_1$  a  $\alpha_3$ ) que serão por sua vez calculados no computador anfitrião via cabo ou via rádio.

Um pequeno resumo das principais características do sistema *CONAC<sup>TM</sup>* é apresentado na seguinte tabela:

	Valor	Unidade
Precisão	1,3, quando aplicado no interior de edifícios, etc. 5, se aplicado no exterior	mm
Custo total do sistema	6000	\$

Tabela 5. Precisão e custo associado ao sistema *CONAC<sup>TM</sup>*.

Uma outra solução dentro desta temática de faróis ativos para o cálculo da posição de um robô móvel, é descrita em [12]. Este sistema desenvolvido pelo Departamento de Controlo Automático da Universidade de Valladolid, é utilizado para determinar a posição e orientação de um robô móvel em ambientes interiores (ou seja, dentro de edifícios). Recorrendo a técnicas de radiofrequência e também a tecnologia laser, o sistema obtém a posição e orientação do robô através de um algoritmo de triangulação. O sistema é constituído por: um laser infravermelho, um motor DC (responsável pela rotação do laser num plano horizontal), um conjunto de faróis fotossensíveis e uma caixa descodificadora de pulsos. O laser, o motor e a caixa descodificadora encontram-se a bordo do robô. O motor ao rodar o feixe do laser ao longo dum plano horizontal consegue desta forma atingir os diferentes faróis fixados nas paredes. Cada farol encontra-se equipada de células fotossensíveis às radiações infravermelhas. Após estas radiações serem recebidas por parte dos faróis, este emite o seu código RF característico (semelhante ao método de operação das *Tags* nos sistemas RFID). A caixa descodificadora recebe este código RF e associa-o ao ângulo do laser obtido por parte do codificador presente no motor. A transmissão de dados é efetuada via do protocolo RS232 para um computador a bordo. Por fim, este computador calcula a posição e orientação do robô através do processo de triangulação.

Na figura seguinte, encontra-se exibido um exemplar do protótipo de farol ativo deste sistema:





Figura 26. Exemplar de um farol do sistema desenvolvido por [12].

Este sistema apresenta o seguinte conjunto de características:

- Efetua 15 leituras de localização por segundo, o que é usualmente suficiente para obter uma localização precisa de robôs moveis que se desloquem a velocidades moderadas;
- Precisão: na ordem de 1 centímetro para questões de posição e  $0,1^\circ$  na orientação;
- Localização pode ser efetuada em grandes áreas de operação recorrendo a um pequeno número de faróis. Por exemplo, o sistema consegue calcular a localização de um robô, de uma forma eficiente, numa área de 450 metros quadrados recorrendo apenas a 3 faróis.

#### 2.4.2.4 NAVEGAÇÃO BASEADA EM MARCOS

Este é um sistema de localização cuja base de funcionamento recai na detecção de **marcos** por parte do robô, ou seja, através da recolha de informação dos sensores a bordo do robô a partir deste diverso leque de objetos e configurações espaciais que funcionam como tal. Estes marcos podem ser formas geométricas (círculos, retângulos ou linhas) que o robô é capaz de reconhecer através dos seus sensores e poderão ou não conter informação (por exemplo um código de barras). Exemplos deste tipo de localização recorrentes a marcos em mapas topológicos para efetuar a localização do robô são empregues em [47], [48], [49] e [50]. Usualmente, estes marcos estão fixos numa posição em relação à qual o robô consegue calcular a sua posição. Antes de se proceder à utilização destes marcos, as suas características devem ser guardadas *a priori* na memória do robô. O objetivo passa por detetar/identificar estes marcos de forma a que se possa assegurar um correto reconhecimento destes mesmos e calcular, de seguida, a posição do robô. É pressuposto que a orientação e posição do robô sejam conhecidas com maior ou menor certeza de modo a que permita ao robô apenas ter de pesquisar por marcos apenas numa área restrita. Devido a estas razões são fundamentais bons pré-requisitos em relação à odometria, para que se obtenha uma precisão elevada.

Na seguinte figura, demonstra-se o princípio de funcionamento deste tipo de sistemas de navegação por marcos:

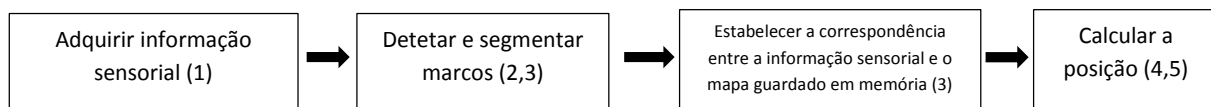


Figura 27: Procedimento normal para navegação por marcos.

Algumas notas que se podem mencionar sobre este tipo de sistemas:

- 1- A informação é adquirida através de faróis e marcos;
- 2- A procura pode ser coagida se se assumir que a estimativa da posição inicial é próxima da posição e orientação verdadeira;
- 3- A deteção dos marcos e o estabelecer a correspondência entre informação sensorial e o mapa guardado em memória são as maiores dificuldades na navegação por marcos (entre elas, o custo e problemas resultantes da sua montagem no terreno);
- 4- Recorrendo a triangulação, a medição do erro é uma função de posição relativa entre o robô e os marcos;
- 5- Através da forma geométrica, a medição do erro é uma função de distância e do ângulo entre o robô e o marco.

São muitos os objetos ou estruturas características que podem funcionar como marcos para possibilitar a navegação por marcos. É possível aglomerá-los em dois grandes grupos: os marcos naturais e os artificiais.

Por marcos naturais podem entender-se por objetos ou estruturas já existentes no ambiente, os quais já possuem uma função característica para lá da navegação em questão, por exemplo navegação por marcos para um robô móvel. Exemplos como portas, paredes, portões e outros objetos verticais podem ser utilizados para se obter a posição do robô, como visto em [49] e em [51]. Ou então o robô *Helpmate* que utiliza as luzes do teto para saber a sua posição pelos corredores do hospital de Danbury [52].

Marcos artificiais são todo um conjunto de objetos ou marcadores específicos que necessitam de ser instalados na área abrangente do robô, permitindo assim a navegação pretendida. Estes marcos podem ser otimizados em termos de tamanho e forma para que melhor se adequem às necessidades do meio ambiente, proporcionando assim uma localização mais eficiente. São vários os exemplos passíveis de se considerarem marcos artificiais, mas provavelmente os mais utilizados sejam os códigos de barras refletoras [53], fita refletora, faróis ultrassônicos ou então padrões que sejam facilmente identificáveis pelo robô, tal como o padrão utilizado por [54] que consiste num conjunto de retângulos negros com pontos brancos.

De seguida, apresenta-se um pequeno resumo das principais características deste tipo de navegação baseada em marcos:

- Marcos naturais oferecem flexibilidade e dispensam modificações no ambiente;
- Marcos artificiais são baratos e permitem ter informação adicional codificada como padrões ou formas de modo a facilitar a navegação;
- A máxima distância efetiva entre robô e marco é substancialmente mais curta que em sistemas de *beacons* ou faróis ativos;
- A precisão na posição depende da distância e do ângulo entre robô e marco. Quanto mais próximo do marco, melhor será a precisão relativa à posição do robô.
- O uso de marcos é consideravelmente mais exigente em termos de processamento computacional quando comparada com a solução baseada em sistemas de faróis ativos;
- Condições climáticas podem afetar o correto reconhecimento da posição dos marcos, podendo originar uma errada estimativa em relação à posição do robô;
- Os marcos necessitam de estar disponíveis no ambiente circundante ao robô;
- Esta solução requer que uma aproximação da posição inicial seja efetuada, de modo a que o robô não perca tempo precioso à procura de marcos para saber onde se encontra. Caso contrário, este processo de procura poderá originar erros na aproximação da posição inicial do robô o que levará a uma interpretação errada dos marcos na área de navegação;
- Uma base de dados relativa aos marcos e sua localização necessita de ser mantida em memória no robô;

### 2.4.2.5 NAVEGAÇÃO BASEADA EM MAPAS (“MAP-MATCHING”)

Posicionamento via Mapas (“*Map-based Positioning*”) também designada por “*Map-Matching*”, é uma técnica em que o robô utiliza a informação recolhida através dos sensores que possui, de forma a recriar um mapa da área envolvente em que se encontra. De seguida, este mapa diga-se “local”, é comparado com outros já previamente guardados na memória do robô ou que possivelmente possam ser fornecidos por outra via [55], [56] e [57]. Se uma correspondência for encontrada então o robô pode computar a sua posição atual e orientação na área geográfica em que irá operar.

Na figura abaixo encontra-se o procedimento usual neste tipo de navegação:

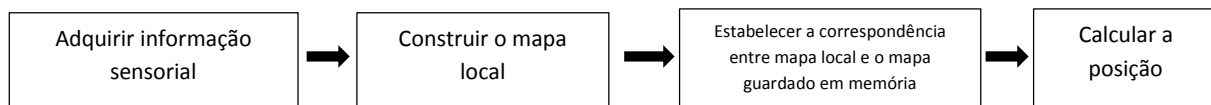


Figura 28: Procedimento geral para navegação baseada em mapas.

Discutam-se agora as vantagens e desvantagens do tipo de sistemas que recorram a esta navegação. Entre as vantagens as seguintes são válidas:

- Esta navegação utiliza estruturas já existentes de modo a estimar/obter a posição do robô sem ter que modificar ou implementar novas estruturas no ambiente em que o robô se encontra;
- Este tipo de cálculo para a posição de um robô poderá ser mais tarde essencial para atualizar mapas relativos ao ambiente que o rodeiam. Estes tipos de mapas são fundamentais para outro tipo de tarefas que irão ser acatadas por outros robôs, como por exemplo evitar obstáculos;
- É uma estratégia que permite ao robô “aprender” um novo ambiente e melhorar a precisão da sua localização através da exploração.

Em relação às desvantagens, cingem-se sobretudo aos requisitos específicos que permitam garantir uma navegação satisfatória. Por exemplo requerem:

- A existência de características fixas distintas o suficiente para o mapeamento;
- O mapa obtido através dos sensores deve ser preciso o suficiente de forma a ter alguma utilidade;
- Uma quantidade considerável de capacidade de processamento e sensorial são necessárias.

Diversas técnicas e estratégias foram expostas e descritas ao longo do capítulo anterior (Cap. 2) de modo a obter de forma eficiente a localização de um objeto móvel. Verificaram-se também os pontos fortes e fracos de cada um dos principais sistemas descritos, podendo concluir que uma forma ideal de obter a posição de um objeto passaria pela junção de técnicas de posição relativa com posição absoluta. Vários métodos foram propostos por, [58], [59], [60], com o objetivo de fundir técnicas, por exemplo, de odometria com medidas de posição absoluta de modo a obter uma posição do objeto com maior fiabilidade.

Um outro exemplo seria empregar técnicas de *dead-reckoning* (odometria e/ou navegação inercial) juntamente com a técnica de GPS ou recorrendo a um sistema de faróis ativos. Um senão provável desta solução poderia ser o seu custo associado e possíveis dificuldades na implementação de toda a infraestrutura necessária no meio. No entanto é facilmente a melhor solução, atualmente, para se obter uma estimativa da localização de um objeto, tanto mais, na área da robótica onde este trabalho encontra o seu propósito.



### 3 SENSORES E O SEU PAPEL NA LOCALIZAÇÃO

Verificou-se que todas as técnicas e respetivos sistemas que fizeram uso dessas mesmas técnicas para obter a localização (e respetiva posição e orientação) de um robô móvel, utilizaram um dispositivo em comum designado por sensor.

No entanto antes que se explique em que consistem e como operam, é necessário endereçar uma distinção entre duas nomenclaturas. Transdutor e sensor podem ser por vezes confundidos e perceptíveis de serem o mesmo dispositivo.

#### 3.1 SENSOR OU TRANSDUTOR

Como descrito em [61], [62], transdutor entende-se por um dispositivo, elétrico ou eletrónico, capaz de converter um sinal numa determinada grandeza física num outro tipo de sinal com diferente grandeza física, com propósito de transferência de informação ou para efeitos de medição. Exemplos possíveis de transdutores passam pelo altifalante (dispositivo que converte os sinais elétricos recebidos na entrada em ondas sonoras à saída) e o microfone (dispositivo que efetua a função contrária ao altifalante, ao converter ondas sonoras recebidas à entrada em sinais elétricos à saída). Um outro usual exemplo para este tipo de dispositivo é o motor de passo (em inglês *motor stepper*). Este motor nada mais é que um motor elétrico *DC* (*Direct Current* – Corrente Direta) sem escovas, capaz de converter sinais de entrada elétricos em mudanças muito precisas de posição de forma rotacional. Ou seja, conforme o ângulo de rotação uma nova posição é-lhe respetivamente atribuída.

Relativamente ao que se entende por um sensor, uma definição possível recai em retratá-lo como um dispositivo que dispõe um sinal de saída característico em resposta a uma quantidade específica de um determinado tipo de sinal de entrada. Este montante de sinal de entrada, que virá a ser medida mais tarde, provoca na saída do sensor um tipo de resposta característico. Este sinal por norma, na maior parte dos sensores modernos, caracteriza-se por ser um sinal de grandeza elétrica (corrente, tensão ou mesmo de

resistência). O que não impede que este mesmo sinal de saída possa também enquadrar-se como sendo relativo a movimento, pressão, fluxo ou um outro tipo de sinal de saída capaz de ser reutilizável num sistema adequado.

Resumindo, o sensor é parte fundamental de um transdutor ao efetuar o *sensing* (sensorização/deteção) da força física em questão e tornando-a passível de reutilização para a forma energética pretendida. Entre os dispositivos mais utilizados encontram-se os seguintes exemplos: codificadores (*encoders*), acelerómetros, giroscópios, bússolas de fluxo, etc.

Sendo a área da robótica a de maior interesse para o objetivo deste documento, um breve estudo sobre os diversos tipos de sensores existentes nesta área pode ter alguma relevância. Portanto, para uma melhor percepção do diverso leque de sensores disponíveis uma breve divisão em categorias é efetuada de seguida.

### 3.2 CATEGORIZAÇÃO E A FORMA COMO SENTEM O MEIO AMBIENTE

Os sensores na área da robótica são possíveis de avaliar de três distintas formas. Inicialmente uma primeira forma que se baseia em apenas descrevê-los conforme a zona de aplicação:

- ***Internos***, que fornecem informação sobre os parâmetros internos do robô, como por exemplo a avaliação da posição ou velocidade de uma roda do robô, ou mesmo o nível atual das baterias, etc. Servem como exemplo destes sensores os potenciômetros, codificadores (*encoders* em inglês), taquímetros e sensores inerciais (incluindo acelerómetros, giroscópios e bússolas);
- ***Externos***, como o nome sugere, facilitam informações sobre fenómenos ocorridos exteriores ao robô no ambiente envolvente. São exemplo destes sensores os de contacto, proximidade, força, ultrassom, visão, laser, infravermelhos e sensores químicos.

Uma segunda forma de classificar os sensores recai na forma como estes efetuam o processo de sensorização do meio, se de uma forma pró-ativa sendo assim denominados por **ativos**, ou se se comportam de uma maneira mais inerte designando-se por isso de **passivos**.

Os ativos medem as alterações efetuadas no ambiente criadas por si ao emitirem quantidades de energia arbitrárias, esperando uma resposta específica conforme o tipo de sinal que emitam. São exemplo de sensores ativos os sensores ultrassónicos, os lasers e os sensores de contacto.

Se apenas receberem energia do ambiente de forma a iniciar a sua operação, denominam-se por passivos. Serve como exemplo de sensor passivo o sensor *PIR* (*Passive InfraRed* – sensor infravermelho passivo) que, como o seu nome indica, se estimulado por ondas de energia infravermelha irá originar uma certa quantidade de carga elétrica a qual é medida pelo módulo no interior do sensor (secção 3.5).

Por último, podem ser caracterizados pelo tipo de grandeza que avaliam aquando do processo de sensorização do meio. Portanto, podem ser considerados: sensores de distância (sensores laser e



ultrassônicos), de posicionamento absoluto – o sistema de GPS ou os codificadores (ou encoders absolutos), sensores ambientais (capazes de indicar a temperatura, humidade) e sensores inerciais (que indicam componentes diferenciais da posição do robô, como por exemplo velocidade ou aceleração, como por exemplo os giroscópios).

Compreendendo agora de uma forma um pouco mais intrínseca, as diversas formas de classificar um sensor, alguns exemplos que retêm especial interesse no âmbito do desenvolvimento deste projeto são sucintamente caracterizados de seguida.

### 3.3 SENSORES INERCIAIS

#### 3.3.1 ACELERÓMETROS

Antes que se clarifique o método de funcionamento de um acelerómetro, é necessário compreender-se o conceito da grandeza física aceleração. Velocidade, posição e aceleração são termos que se encontram interligados, pois a velocidade é a primeira derivada da posição e a aceleração a segunda derivada. No entanto, obter-se a posição através de derivação em ambientes propícios a elevadas quantidades de ruído pode traduzir-se em elevados erros mesmo utilizando circuitos complexos e sofisticados de acondicionamento de sinal. Mas como o cientista Isaac Newton provou com a sua 2ª lei do movimento ( $a = \frac{\sum F}{m}$ ) [63], a aceleração de um objeto é dada pela quantidade de força necessária para mover cada unidade de massa.

Portanto, ao contrário do que se possa pensar, os acelerómetros não obtêm a aceleração derivando os valores obtidos de velocidade ao longo do tempo de um objeto, mas sim medindo as variações desta grandeza física denominada por força. Geralmente, detetam estas alterações de força conforme uma massa (intrínseca ao acelerómetro, denominada usualmente por massa inercial) seja comprimida ou de cada vez que uma força atue sobre ela. São vários os tipos de acelerómetros disponíveis, entre eles os:

- **Mecânicos** – versões mais simples, constituídas por uma massa que se encontra presa a uma mola suspensa colocada fora do encapsulamento do acelerómetro. Assim que o acelerómetro inicia o seu movimento, esta massa fica para trás e a mola estica com uma força respetiva a qual corresponde a um determinado valor de aceleração. Sismógrafos são exemplos de acelerómetros mecânicos. Cada vez que um sismo ocorre uma caneta suspensa anexada a grandes massas deteta as variações das forças sísmicas de forma a calcular mais tarde a intensidade do sismo. O princípio de funcionamento de um acelerómetro mecânico de forma muito simples pode ser observado na seguinte figura:

## Mechanical accelerometer

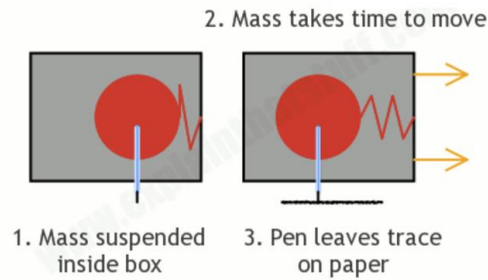


Figura 29. Princípio de funcionamento de um acelerómetro mecânico [13].

- **Capacitivos** – Anteriormente verificou-se que uma caneta registava as alterações de força sofrida pelo acelerómetro em papel, por exemplo um sismógrafo. No entanto, esta caneta pode ser substituída por sinais elétricos ou magnéticos. É neste cenário que entram os acelerómetros capacitivos, que utilizam condensadores, mais concretamente a distancia entre as placas que os constituem, para obter a aceleração.

## Capacitive accelerometer

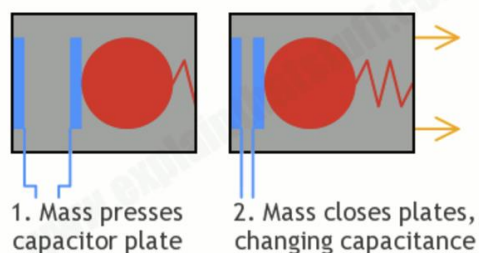


Figura 30. Acelerómetro Capacitivo [13].

A massa constituinte do acelerómetro ao pressionar as placas do condensador provoca alterações na distância entre as duas placas do condensador. Estas diferenças observadas na distância entre placas permitem obter a capacitância do condensador, o que por consequente permite obter o valor da força atuante e mais tarde a aceleração.

- **Piezoelétricos** – Por vezes, alguns acelerómetros utilizam cristais piezoelétricos como o elemento que interage com a massa que sofre as todas as forças resultantes do movimento. Cada vez que o acelerómetro se move, a massa aperta o cristal e gera uma pequena tensão elétrica proporcional à aceleração.

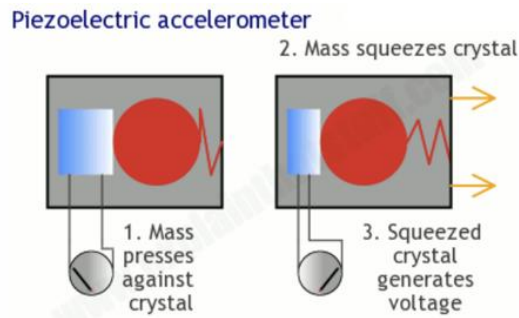


Figura 31. Acelerómetro Piezoelétrico [13].

Resumindo, um acelerómetro nada mais é que um sensor que traduz o deslocamento efetuado pela massa inercial relativamente à proteção ou encapsulamento do acelerómetro através da produção de algum tipo de sinal específico como resposta a este deslocamento. Este tipo de sinal, usualmente de carácter elétrico, é depois utilizado como sendo função da aceleração.

Como dito em [14], os acelerómetros são dispositivos muito robustos e estanques. Quer isto dizer que cooperam de forma ideal com as exigentes condições climáticas de algumas zonas geográficas, como por exemplo em Aveiro, em que a elevada humidade é um fator a ter em conta. Costumam aplicar-se também em locais fabris, em que o ruído e vibrações gerados pelos diversos motores são de alguma forma usual, pois são pouco sensíveis a estas contrariedades.

### 3.3.2 GIROSCÓPIOS

Como dito em [64], giroscópios representam um papel fundamental na obtenção da posição de um robô móvel pois conseguem conciliar os problemas resultantes de uma abordagem que recorra apenas a técnicas de odometria. Como foi dito anteriormente, na secção 2.4.1, um dos problemas associados a uma posição baseada em odometria recai na integração dos erros ao longo do tempo oriundos de pequenas alterações na orientação do robô. Com a utilização de sensores como o giroscópio, tais erros são possíveis de ser detetados e corrigidos de imediato. São dispositivos que não são afetados pela força gravitacional da Terra ao contrário dos acelerómetros, sendo por isso um bom complemento aos acelerómetros aquando na obtenção da posição de um robô móvel.

Giroscópios são dispositivos que fazem uso da força gravitacional da Terra para o seu funcionamento e deste modo determinarem a orientação de um objeto ao medirem velocidade angular. Velocidade este normalmente medida em graus por segundo ( $^{\circ}/s$ ) ou rotações por minuto (RPM).

Um giroscópio **mecânico** tem como princípio fundamental a conservação do momento angular: *“Em qualquer sistema de partículas, a totalidade do momento angular de um sistema relativamente a qualquer ponto fixo no espaço mantem-se constante, atendendo a que nenhuma força externa atue no sistema.”*[14].

Um giroscópio mecânico é constituído por um enorme disco (**rotor**) que roda em torno de um eixo, eixo esse que se encontra enclausurado numa armação livre de rodar em torno de um ou dois eixos.

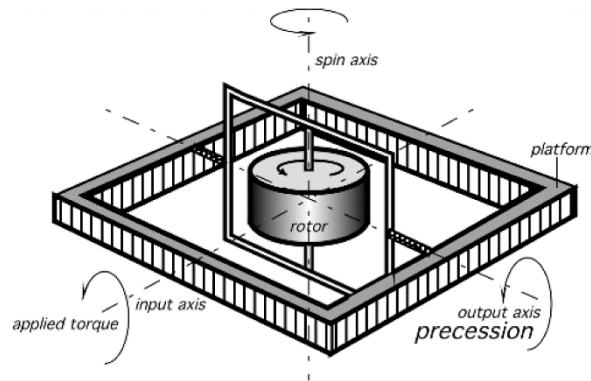


Figura 32. Giroscópio mecânico[14].

O rotor ao rodar livremente, tende a preservar o seu momento angular. A plataforma do giroscópio ao girar em torno do eixo de entrada, o giroscópio gera uma força de rotação perpendicular ao eixo de saída (também designada por binário ou torque), girando assim o eixo de rotação (“*spin axis*” da Figura 32) em torno do eixo de saída. Este fenómeno designado por precessão e baseia-se na conservação do momento angular de um objeto conforme o seu eixo de rotação é alterado. Uma forma de precessão ocorre, por exemplo, quando um pião gira sobre uma superfície. O pião gira sobre o seu próprio eixo, mas este eixo oscila ligeiramente.

As possíveis fricções com origem na rotação do rotor e o decaimento de qualidade do material que funciona como suporte de toda a estrutura constituinte do giroscópio mecânico, são duas das principais razões por que este tipo de giroscópios caiu em desuso e novas soluções foram desenvolvidas que não recorressem a materiais físicos para o cálculo da velocidade angular.

É aqui que surgem os giroscópios **óticos**, dispositivos constituídos por um número reduzido ou mesmo nulo de partes móveis e que fazem uso do Efeito de Sagnac para o seu princípio de funcionamento.

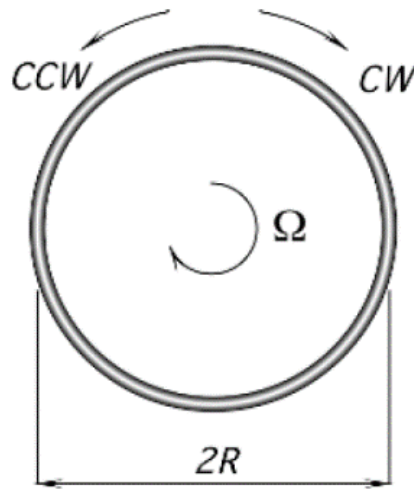


Figura 33. Efeito de Sagnac num anel óptico de raio  $R$  [15].

Como é possível observar através da Figura 33, um feixe de luz irradiado por um laser é dividido em dois feixes distintos ao longo de um anel óptico. Um feixe propaga-se no sentido dos ponteiros do relógio (*CW-Clockwise* na figura) e o outro no sentido contrário (*CCW – Counterclockwise* na figura). As interferências construtivas e destrutivas que são obtidas a partir de divisões ou junções de secções dos dois feixes podem ser utilizadas para se determinar a direção de rotação do dispositivo. Exemplos de giroscópios óticos que seguem este princípio, o efeito de Sagnac são: o Giroscópio Anel Laser (*Ring Laser Gyroscope* ou **RLG**) e Giroscópio Fibras Ótica (*Fiber Optic Gyroscope* ou **FOG**). Num RLG o anel é parte do laser, como em [65], enquanto que num FOG a luz originária de um laser externo é fornecida via cabo de fibra ótica. Exemplos destes dois tipos de giroscópios são apresentados nas seguintes figuras:

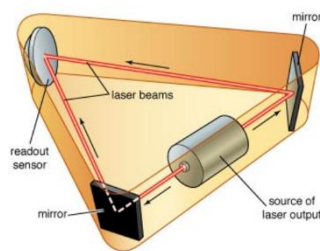


Figura 34. Giroscópio Anel Laser (RLG) [16].

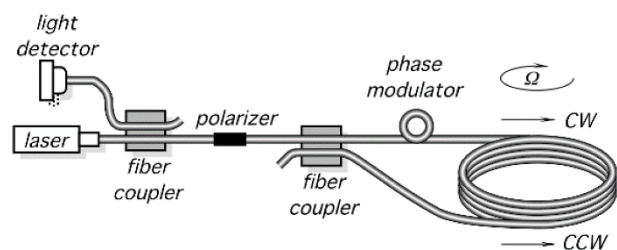


Figura 35. Giroscópio fibra ótica (FOG) [14].

Através da Figura 34, consegue-se constatar como o laser é parte integrante do RLG, enquanto que na Figura 35 é bem visível o efeito de Sagnac na bobina de fibra ótica criando os dois feixes pretendidos (um no sentido CW e outro no sentido CCW).

Contudo com o avançar da tecnologia a necessidade de se desenvolver uma solução de baixo custo e de alguma forma compacta intensificou-se. Isto levou à criação dos giroscópios *MEMS* (*micro electro-mechanical systems*). São de tamanho reduzido sendo assim capazes de serem integrados em chips e utilizam elementos mecânicos vibratórios de forma a detetar rotação. Qualquer corpo vibratório tem uma tendência de continuar a vibrar ao longo de um plano. Qualquer alteração na orientação da plataforma à qual se encontra anexado o corpo vibratório exerce uma força nessa mesma plataforma. Esta força pode ser mais tarde medida e utilizada para calcular a rotação. Exemplos de giroscópios MEMS podem ser mais detalhadamente analisados em [66].

Em jeito de conclusão, as principais vantagens dos giroscópios MEMS quando comparados com os do tipo RLG ou FOG são:

- São espaço-eficientes – ao estarem disponíveis na forma de chips, podem ser embutidos em circuitos eletrónicos;
- Não possuem componentes móveis como os RLG ou FOG e desta forma não necessitam de manutenção;
- Encontram-se disponíveis a preços muito mais baixos que os dos tipos RLG ou FOG.

### 3.4 ENCODERS (CODIFICADORES)

Codificadores (ou *encoders* na terminologia anglo-saxónica) são transdutores capazes de traduzir uma quantia mecânica numa outra de grandeza elétrica, através da leitura de um padrão codificado presente num disco rotacional (caso sejam do tipo rotacional) ou numa régua (se são do tipo linear). *Encoders* são classificados de acordo com estes três parâmetros:

- Método utilizado para a leitura do elemento codificado – se de contacto ou não-contacto;
- O tipo de sinal de saída – se é uma palavra digital absoluta ou se é uma serie de impulsos incrementais;
- Qual o fenómeno físico utilizado para obter o sinal de saída – se através de condução elétrica, magnetismo, ótico ou capacitivo.

O seu uso em robótica móvel encontra aplicação aquando a velocidade angular de um eixo em rotação, por exemplo de um motor ou de uma das rodas do robô, é transformada/convertida em impulsos eletrónicos (caso seja do tipo incremental) ou numa palavra digital codificada específica (caso seja do tipo absoluto) para a determinação da posição do mesmo. Vários exemplos como [67], [68] por exemplo, empregam este tipo de dispositivos para a obtenção da posição de um robô móvel.

O tipo de *encoder* utilizado com maior frequência na área da robótica passa pelo *encoder* que faz uso do fenómeno da ótica (o *encoder* ótico), o qual encontra inicialmente empregabilidade por volta do início dos anos 40 [69]. O seu princípio de funcionamento é bastante simples, recorrendo a um par de discos com segmentos opacos e transparentes e a uma fonte de luz colocada de um lado, como por exemplo um LED – (*Light Emitting Diode* ou díodo emissor de luz), e um fotossensor, capaz de detetar a luz irradiada pelo laser do outro, consegue descrever a posição angular do eixo de rotação de uma forma incremental ou absoluta. É aqui que surgem os *encoders incrementais* e os *encoders absolutos*, categorizados de acordo com o seu tipo de sinal de saída.

#### 3.4.1 ENCODERS ÓTICOS INCREMENTAIS

Este tipo de *encoders* emitem uma sequência de pulsos por cada revolução efetuada pelo eixo onde se encontram aplicados. Este número de impulsos indica que distância (angular ou linear) foi percorrida, mais tarde circuito eletrónico auxiliar capaz de contar estes impulsos gerados calcula a distancia total percorrida pelo dispositivo em relação a um ponto de referência. Este ponto de referência, tem um papel fundamental neste tipo de dispositivos pois todo o processo de deslocamento é tido em conta a distância a que este ponto

se encontra a cada nova incrementação. Esta é também uma das principais desvantagens deste tipo de *encoders*, sempre que se quiser iniciar o movimento do robô móvel, este tem de ser deslocado primeiramente até este ponto de referência. Esta situação é também válida para situações em que se desliga e se volta a ligar o *encoder*, pois a contagem é perdida após cada período de utilização.

Um exemplo de um *encoder* incremental é apresentado de seguida:

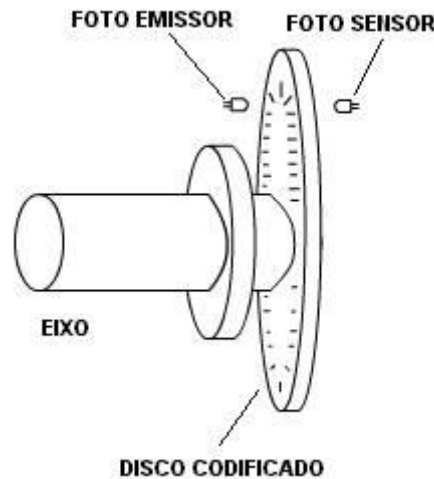


Figura 36. *Encoder* Incremental [17].

O disco apresentado na figura anterior contém ranhuras e de cada vez que a deteção de luz entre o emissor e o recetor passa de contínua para interrompida, uma série de impulsos elétricos “1” (nível lógico HIGH) e “0” (nível lógico LOW) são gerados. Contando estas transições de HIGH para LOW consegue-se obter a deslocação angular que o eixo sofreu e dessa forma saber que valor respetivo de distância se avançou em relação ao ponto de referência. A contagem destas transições entre os diferentes estados lógicos está diretamente relacionada com o conceito de resolução.

**Resolução** pode ser definida como sendo a quantia mais pequena que provoca uma alteração de estado lógico na saída. Para se entender melhor este conceito, um *encoder* incremental de canal simples (apenas gera uma sequência de impulsos na saída) é ilustrado na figura seguinte:



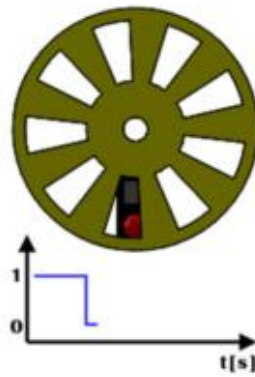


Figura 37. *Encoder* incremental e respetivo sinal de saída [18].

Da figura 35 observa-se que o disco codificador do *encoder* apresenta 9 ranhuras. Portanto, neste *encoder* ocorrem 9 transições de valor lógico “1” para “0” numa rotação completa deste disco e a sua resolução toma o valor de:  $360^\circ/9 = 40^\circ$ . Sendo assim  $40^\circ$  é o menor valor com que se conseguem diferenciar as transições entre região opaca e ranhura do disco, e é por sua vez também o período da onda visível na figura. Com esta configuração de canal simples, o *encoder* não é capaz de identificar em que sentido o movimento se efetua. Isto é ultrapassado através de uso do fenómeno de quadratura, que consiste em nada mais que atrasar uma onda em relação à outra em  $90^\circ$ , ou seja, um quarto de período da onda. Este processo permite não só, como já discutido, descobrir qual o sentido do movimento como também proporciona um aumento na resolução do *encoder*. Mais detalhe de como este processo é efetuado pode ser encontrado em [18].

### 3.4.2 ENCODERS ÓTICOS ABSOLUTOS

Este tipo de dispositivo, ao contrário dos incrementais, codifica cada posição angular do eixo numa palavra digital única. O processo de funcionamento é em tudo semelhante ao *encoder* incremental, ao utilizar novamente dois discos – um fixo e um outro móvel (estando este codificado), uma fonte de luz de um lado e um fotossensor recetor do outro. Neste tipo de transdutor, um feixe de luz irradia a totalidade do disco codificador (fixo e solidário com o eixo de rotação), enquanto que o disco fixo apresenta apenas uma ranhura radial num plano paralelo ao do disco codificado. Desta forma, o feixe de luz ilumina uma ranhura codificada por cada posição angular, detetando assim diferentes padrões para diferentes posições angulares, obtendo-se assim a posição. O método mais característico para a codificação dos discos passa pela codificação de Gray, binário natural ou binário decimal [70].

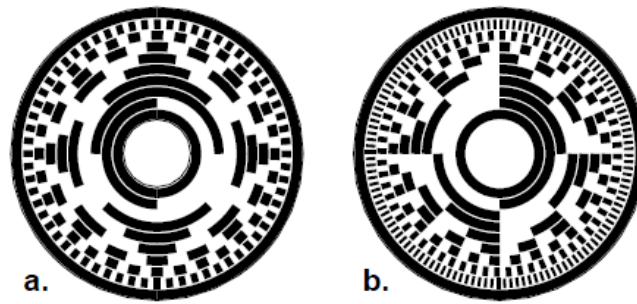


Figura 38. *Encoder* Absoluto com dois tipos de codificação [19].

a. – Codificação Gray b.– Codificação binária natural

Na figura 38 apresentam-se as diferenças entre um disco codificador baseado em codificação Gray e uma codificação binária natural. Como é visível uma vantagem da codificação Gray em relação à binária resulta na alteração de apenas um bit entre secções, uma possível vantagem em eliminar ambiguidades assíncronas possivelmente criadas pelos componentes eletrónicos e mecânicos do *encoder*.

Este tipo de *encoders* apresentam uma grande vantagem quando comparados com os incrementais, que recai no facto de a codificação de todo o disco ser um fator conhecido *à priori*. Portanto, mesmo que a energia seja interrompida a meio do movimento, o funcionamento do *encoder* não sofre alterações. Não necessitam assim de uma referência inicial para ter conhecimento da posição angular do eixo em cada rotação, podendo ser desta forma transdutores úteis em situações em que potenciais percas de pontos de referência oriundos pela interrupção de energia.

### 3.5 SENSORES PIR E IR BREAKBEAM

Sensores Passivos de Radiação Infravermelha ou PIR (*PIR – Passive Infrared*), são dispositivos que permitem detetar movimento através da radiação infravermelha. Tornaram-se extremamente populares para a segurança e manutenção de energia de sistemas. O elemento sensorial do PIR deve responder a radiação infravermelha no espectro eletromagnético com comprimento de onda compreendido entre 4 a 20 micrómetros [14, p. 245]. O elemento sensorial escolhido para a deteção passa por um sensor piroelétrico, responsável pela deteção de radiação infravermelha.

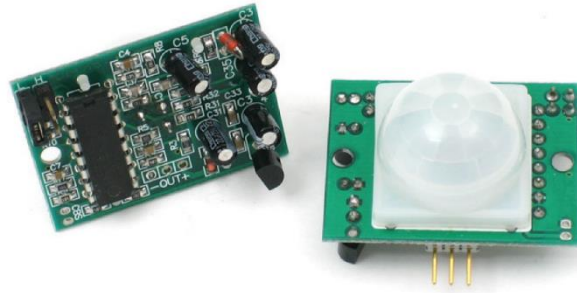


Figura 39: Sensor Passivo infravermelho. Vista traseira e frontal, respetivamente [20].

Sendo de tamanho reduzido, de baixo custo e de consumo energético pouco exigente, tornaram-se num elemento muito utilizado para a deteção de movimento em locais de trabalho e/ou em residências. O sensor num detetor de movimento encontra-se dividido em duas metades constituídas por material sensível a infravermelhos, variando o sinal de saída conforme uma das metades detete uma maior ou menor quantidade de radiação infravermelha. O processo de deteção deste tipo de sensor passivo é algo de muito rudimentar.

Essencialmente divide-se em dois estados, um em que se encontra inativo (as duas partes detetam a mesma quantidade de radiação infravermelha, oriunda das paredes ou do meio exterior), e um outro em que um corpo quente, tal como um animal ou ser humano, atravessa o campo de visão do sensor. Com a ajuda da seguinte imagem torna-se fácil de perceber o funcionamento do sensor aquando da passagem de um corpo quente pelas duas partes sensíveis a radiação infravermelha de um sensor PIR.

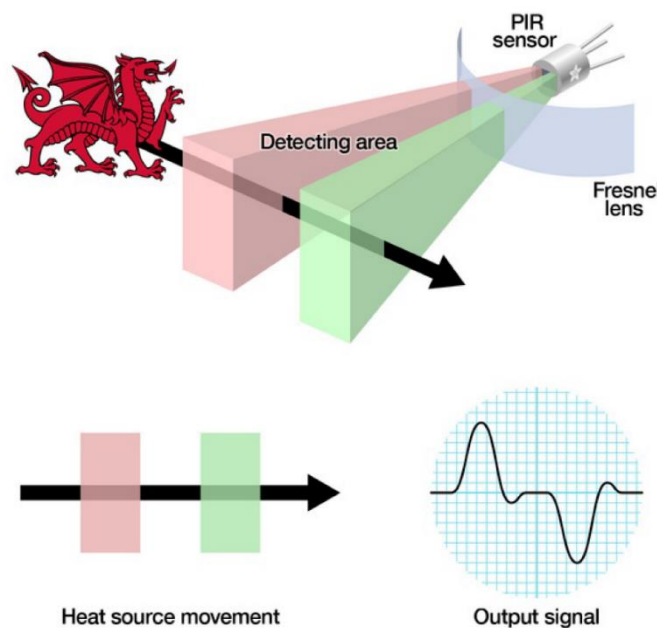


Figura 40: Processo de deteção de um corpo quente por parte de um sensor PIR [20].

Como é possível observar na figura 40, cada vez que um corpo que emane radiação no espectro do infravermelho passe pelo sensor PIR, atravessa inicialmente uma das duas metades do sensor originando uma diferença positiva no sinal de saída. O inverso sucede quando o corpo deixa a área de sensoriamento, sendo assim gerada uma diferença negativa no sinal de saída. Estas variações nos pulsos digitais são o que o sensor PIR é capaz de fornecer. Para além do elemento sensorial piroelétrico, um detetor de infravermelhos necessita também de um dispositivo ótico capaz de focar o objeto em questão, melhorando assim a sensibilidade na área de deteção do sensor. A escolha das lentes de plástico Fresnel tornaram-se cada vez mais populares pois são baratas, podem ser moldadas na forma pretendida e para além da funcionalidade de foco, atuam como janelas, ao protegerem o interior do detetor do exterior contra poluentes e humidade [14, p. 137].

Na tabela seguinte reúnem-se um conjunto de caraterísticas principais deste tipo de sensores passivos infravermelhos.

	<i>Valor</i>	<i>Unidade</i>
<i>Sensibilidade no alcance</i>	Até 7m ( na direção de um cone de 120 graus)	metros
<i>Alimentação</i>	5 a 12	V(Volts)
<i>Sinal de saída digital</i>	Estado lógico High (3V) ou estado lógico Low (0V)	V(Volts)
<i>Preço</i>	10	\$(dólares)

Tabela 6. Resumo das caraterísticas de um PIR da adafruit [22].

É dentro desta temática de deteção de movimento a partir de radiação infravermelha, que um sensor mais simples e mais rápido que o sensor PIR encontra empregabilidade, o **IR (InfraRed) BreakBeam Sensor** (Sensor Infravermelho Quebra-Feixe)[21]. Um conjunto de dois sensores deste tipo, um emissor e um recetor são necessários. O emissor emite um feixe de luz infravermelha contínuo, o qual vai ser recebido pelo sensor recetor sensível a este tipo de luz. Sempre que algo quebre este feixe (daí a origem do seu nome) se interponha entre o emissor e o recetor, o recetor informa o utilizador de que tal aconteceu através de um sinal digital. Uma versão de 5 mm e outra de 3 mm encontram-se disponíveis, conforme a distância a que se pretenda afastar o emissor do recetor mantendo uma leitura correta.

Um exemplar deste tipo de sensores encontra-se ilustrado de seguida:



Figura 41. Sensores Infravermelhos Quebra Feixe. À esquerda da figura o recetor e à direita o transmissor [21].

De realçar da figura 41, o cabo de cor branca no recetor que representa a saída digital que oscila entre valores lógicos HIGH (estado lógico 1 – não há quebra do feixe) ou LOW (estado lógico 0 – o feixe encontra-se quebrado).

Concluindo, apresentam-se as principais características deste conjunto na tabela seguinte.

	Valor	Unidade
<b>Alcance</b>	25 ou 50 cm	cm(centímetros)
<b>Alimentação</b>	3,3 ou 5 V	V(Volts)
<b>Corrente Exigida pelo Emissor</b>	10 mA @ 3.3V ou 20 mA @ 5 V	mA(miliAmpere) @ Volts
<b>Corrente de Saída máxima no recetor</b>	100 mA	mA(miliAmpere)
<b>Tempo de resposta</b>	20 ms	ms(milissegundos)

Tabela 7. Principais características de um conjunto *IRBreakBeam* [21].



## **4 SOLUÇÃO PROPOSTA**

### **4.1 ESTRUTURA E IDEIA**

No terraço do Instituto de Telecomunicações (IT) de Aveiro encontra-se instalado um carril metálico com um formato circular apoiado por inúmeros pilares, também eles metálicos. Esta infraestrutura foi instalada para suportar uma carga móvel que se deslocará ao longo deste carril para efeitos futuros de testes de mobilidade na rede de testes já existente (o AMAzING). Na seguinte figura é possível visualizar a estrutura metálica montada no terraço e também a carga que percorrerá o carril a diferentes velocidades.



Figura 42. Estrutura montada no terraço do IT.

A escolha do veículo para a carga móvel recaiu na opção de uma roda de uma bicicleta elétrica embutida num sistema metálico que permitisse a circulação ao longo do carril em relativa segurança. Das seguintes figuras (43 e 44), é possível constatar a presença da roda e nomeadamente o motor elétrico, o conjunto de baterias que perfazem um total de 60V e uma caixa de cor branca onde se encontram o controlador e, respetivamente, um conjunto de sinais elétricos capazes de serem utilizados para fins de controlo sobre o motor.

Um esquemático de todos os componentes dentro desta caixa de controlo, encontra-se apresentado em anexo (Anexo A) para uma melhor compreensão de que sinais de controlo e de potência se encontram disponíveis com este protótipo.





Figura 43. Veículo escolhido como carga.



Figura 44. Caixa de controlo do veículo.

O propósito deste documento, passa pela elaboração de um sistema de localização capaz de fornecer a posição da carga ao longo deste carril. Para tal, duas entidades são fulcrais de serem designadas para uma melhor compreensão do seu funcionamento. Uma sendo a **carga/protótipo** que se irá ligar a uma outra, o **servidor**, via um dos inúmeros meios de comunicação possível para a interligação entre dois elementos, como por exemplo, *WiFi*, *Bluetooth*, *RF*, *XBee*, etc. O meio de comunicação escolhido no desenvolvimento deste projeto para a troca de informação entre as duas entidades recai no *WiFi*. O protocolo de comunicação (o tipo de mensagens trocadas entre os dois) utilizado para a troca de informação e de que forma é implementado descreve-se na próxima secção (Cap. 4.2).

Portanto, estando o meio de comunicação entre as duas partes do sistema resumido de forma sucinta, resta apenas descrever que função tem cada uma no sistema, para se obter uma estimativa da posição da carga.

Dois pares de sensores IR BreakBeam são colocados na carga, em sítios estratégicos (ver Figura 45), de modo a detetarem corretamente a ocasião em que haja um cruzamento de um pilar de apoio do carril. Cada vez que esta situação ocorre, o feixe de cada par de sensores infravermelho deteta uma quebra e esta informação é comunicada via *WiFi* ao servidor. A empregabilidade de dois pares de sensores deve-se à necessidade de se pretender distinguir qual a direção do sentido do movimento da carga, se este se efetua no sentido dos ponteiros do relógio (*CW* – *Clockwise*) ou num sentido contrário aos ponteiros do relógio (*CCW* – *Counterclockwise*).

Nomeadamente às funções do servidor, recaem essencialmente em duas. Uma primeira em que consiste na criação de um ponto de acesso (*AP* – *Access Point*), podendo assim a carga ligar-se à rede *WiFi* criada pelo servidor. Em relação à segunda, consiste em todo o acondicionamento de informação enviada pela carga de modo a se identificar corretamente o pilar metálico em que a carga se encontra. Para além destas funcionalidades, o servidor é também capaz de detetar: situações em que a carga pare num determinado ponto do carril ou mesmo sobre um pilar, e o caso em que ocorram inversões no sentido de movimento.

A figura seguinte ilustra a forma como a carga deteta um cruzamento de um pilar do carril.

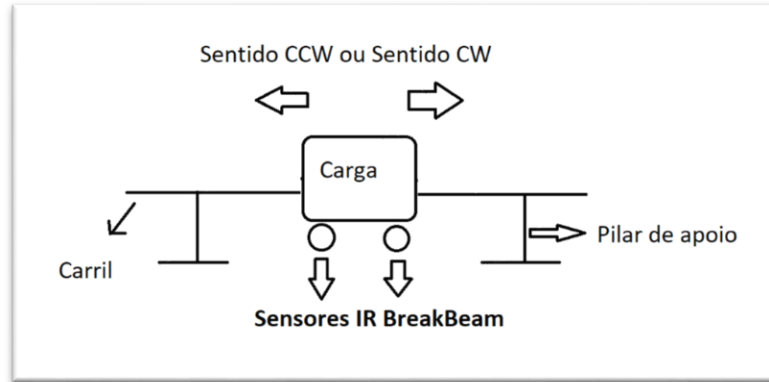


Figura 45. Esquema simples do método de detecção de pilares por parte do sistema.

Após cada cruzamento de pilar, cada sensor mantém uma contagem de pilares que detetou. Ora, para que esta contagem seja feita de forma eficaz, um pilar que sirva de referência (pilar número 0) é escolhido entre os inúmeros pilares de apoio existentes. Portanto, inicialmente a carga é colocada neste pilar e a contagem é inicializada ao longo do carril, de uma forma incremental caso o movimento seja no sentido CW ou caso seja CCW o número de cada pilar decrementa após cada passagem da carga. Este pilar de referência indica também cada vez que uma volta completa ao carril é efetuada de forma a que os sensores não contem indefinidamente, mas sim até a um número igual ao total de pilares existentes. A estrutura montada no terraço do IT possui um total de 40 pilares de suporte. Sendo assim, a numeração dos pilares inicia-se em 0 no pilar escolhido como referência em 0 e prolonga-se até um máximo de 39.

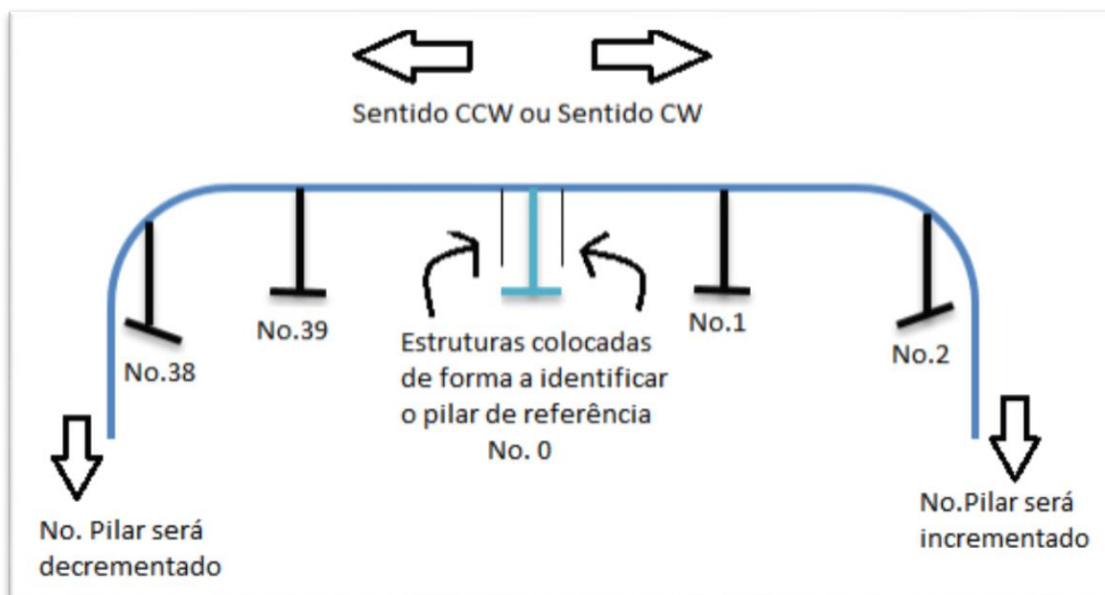


Figura 46. Pilar de referência (a azul ciano) e mapeamento dos pilares.

Na figura 46, é possível constatar que duas estruturas auxiliares são colocadas (uma antes e outra depois do pilar de referência) de forma a distinguir este pilar dos outros. Exemplifica-se também o método de como a contagem dos pilares é efetuada, de uma forma incremental caso o movimento seja no sentido CW e de uma forma decremental caso este seja no sentido CCW.

Concluindo, apenas falta enquadrar esta solução numa das categorias descritas anteriormente neste documento ao longo de todo o capítulo 2. Este sistema calcula uma estimativa para a posição de uma carga ao longo do carril, **relativamente** a um ponto de referência pré-estabelecido. Por esta razão, a localização efetuada através deste sistema pode ser enquadrada como sendo uma localização relativa.

## 4.2 IMPLEMENTAÇÃO DA IDEIA

Verificou-se que o meio de comunicação escolhido para a transmissão de dados entre as duas partes constituintes do sistema seria via WiFi. No entanto, primeiro é necessário estipular que dispositivo permite este tipo de comunicação.

O **ESP8266** é um microcontrolador desenvolvido pela *Espressif Systems* que já tem embutido nele um módulo WiFi que permite efetuar ligações TCP até a um máximo de 5. Existe um leque diverso dentro da família ESP8266, desde o ESP-1 (a versão de origem) até ao mais recente (no momento em que se escreve este documento) e utilizado neste projeto o ESP-12E. As diferenças entre as várias versões lançadas do microcontrolador ESP8266 não passam pelo microprocessador que se mantém inalterado em todas elas, mas sim pelo número de pinos que o módulo disponibiliza ao utilizador. Por exemplo, o ESP-1 tem um total de 8 pinos enquanto que o ESP-12E apresenta um total de 22 pinos. Este aumento no número de pinos, permite um maior numero de entradas/saídas digitais como também a possibilidade de transmissão de informação via série através de uma forma assíncrona (via o protocolo de comunicação UART) ou então de forma síncrona (via protocolo SPI). No mercado, existem inúmeras placas de desenvolvimento (*development boards*) onde o microcontrolador ESP8266 é montado. A *board* escolhida para este projeto é a *NodeMCU devKit* que disponibiliza ao utilizador a versão ESP-12E.

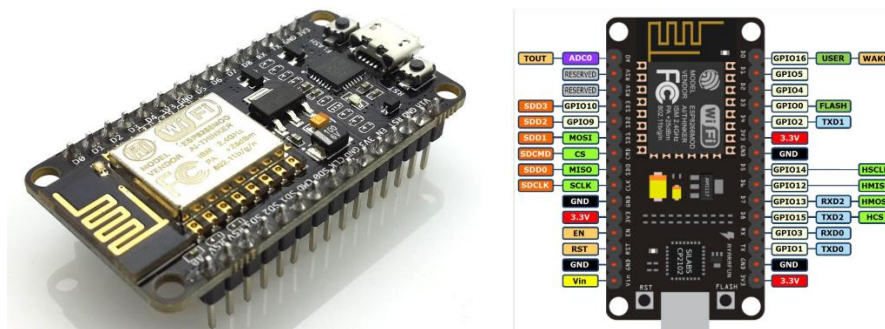


Figura 47. *NodeMCU devKit board* e a disposição dos diferentes pinos da *board*.

Na figura 47 para além dos pinos de entrada e saída (GPIO de 1 a 16), pinos de alimentação, a antena embutida, uma entrada micro-USB, etc. é também possível constatar os 22 pinos do módulo WiFi ESP-12E. De seguida, apresentam-se as principais características desta board na tabela seguinte:

Alimentação	3,3V(no entanto tolera valores entre 3.0 - 3,6 V)
Corrente consumida	10µA a 500mA (64 mA em média)
Processador	Tensilica L106 de 32 bits
Clock (relógio) processador	80 MHz
Memória	Memória flash de 4MB
GPIO (Pinos entrada/saída)	17 (multiplexados com outras funções)
WiFi	802.11 suporta b/g/n/d/e/i/k/r
ADC	1 de 10 bits
Ligações TCP	Máximo de 5 ao mesmo tempo

Tabela 8. Principais características da board NodeMCU devKit.

Portanto, tendo agora resumidas as características da unidade de processamento que será fulcral para a transmissão da informação captada pelos sensores *BreakBeam* do lado da carga para o lado do servidor, a implementação da solução proposta pode ser exposta.

Uma *board NodeMCU* do lado da carga recebe via pinos digitais GPIO (D1 e D2 – GPIO5 e GPIO4 respetivamente) recebem os sinais lógicos enviados pelos dois pares de sensores *BreakBeam* (estado lógico ‘1’- o feixe infravermelho entre sensores é contínuo ou estado lógico ‘0’- o feixe foi quebrado).

Entretanto uma segunda *board NodeMCU*, cria um ponto de acesso (AP) *WiFi* no qual a board do lado da carga se irá ligar estabelecendo assim uma ponte via *WiFi*. O protocolo de comunicação escolhido para a troca de mensagens entre o par de placas recaiu no tipo de mensagens que se designam por WebSockets. WebSockets são simples ligações TCP entre o cliente (carga/carrinho) e o servidor que permitem uma transmissão bidirecional, *full duplex* e de reduzida complexidade o que resulta numa conexão de baixa latência. No lado do servidor abre-se uma porta TCP (neste projeto a porta utilizada foi a porta 81) na qual o servidor fica à escuta de mensagens WebSockets enviadas pelo cliente. O servidor é depois capaz de implementar uma máquina de estados que divide em diferentes grupos as diversas situações que o carrinho/carga experiencia ao longo do seu movimento pelo carril. Podem ser resumidos em 5:

0. Estado em que se informa qual o último pilar cruzado e em que sentido.
1. Situação em que se cruzou um pilar no sentido CW ou CCW.
2. Cenário em que ocorreu uma inversão no sentido do movimento (de CW para CCW ou vice-versa, após a passagem de um pilar).
3. Carga parou sobre um determinado pilar.
4. O movimento da carga cessou durante mais que um tempo limite entre um par de pilares.

O módulo WiFi quando “adormecido” consome apenas 10  $\mu\text{A}$ , mas este consumo pode sofrer um pico até um máximo de 500mA quando se transmite informação via WiFi. Logo, uma fonte de alimentação bem regulada é necessária para o bom comportamento do sistema. De seguida expõe-se como se projetou a energia necessária para o módulo WiFi do lado da carga conseguir transmitir de forma correta a sua informação.

Como pode ser observado pelo Anexo A, existe um conversor de (48/60 para 12) Volts no interior da caixa de controlo do protótipo/carga. Esta tensão DC pode ser utilizada para alimentar todos os componentes necessários do lado da carga, entre eles a board NodeMCU (e respetivo módulo WiFi) e os sensores BreakBeam. Na tabela 7, verificou-se que os sensores BreakBeam tanto podem ser alimentados a 3.3 V como a 5 V, conforme o alcance que se pretenda. Optou-se por alimentação a 5V de modo a que não haja problemas em relação à leitura entre recetor e transmissor do par de sensores infravermelhos. A board NodeMCU, ver figura 45, possui uma entrada “Vin” que permite tensões de alimentação entre 5 a 10V. Sendo assim, optou-se por uma alimentação a 9V através de um regulador de tensão (LM7809) capaz de fornecer 9V e até um máximo de 1A como corrente de saída (relembre-se a exigência do módulo WiFi).

Um esquemático de tudo isto é exibido de maneira a tornar mais fácil a compreensão do que se acabou de descrever.

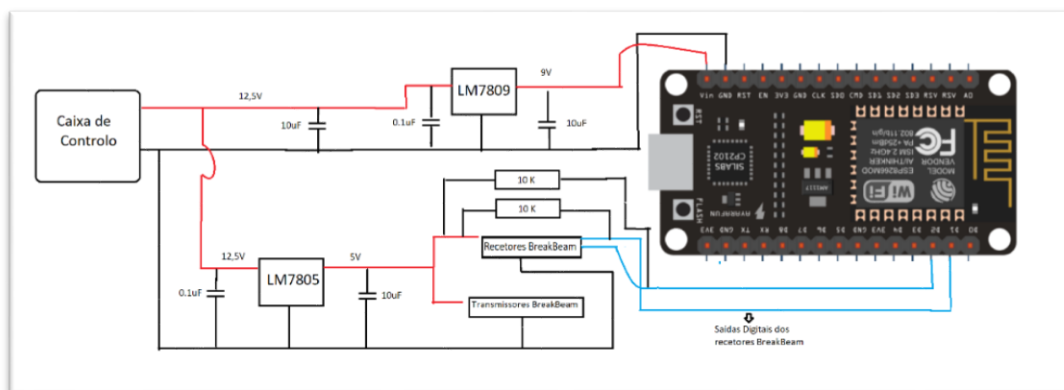


Figura 48. Alimentação do NodeMCU (lado Carga) e dos Sensores BreakBeam.

Encontrando-se agora a placa NodeMCU do lado da carga corretamente alimentada, pode-se proceder à comunicação via WiFi trocando mensagens WebSocket com o servidor da seguinte forma:

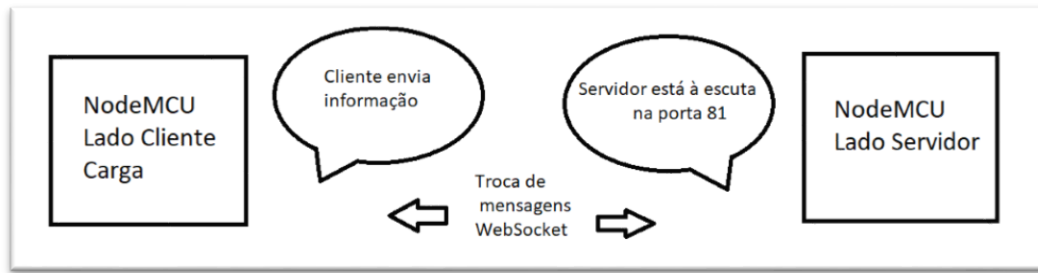


Figura 49. Comunicação entre Cliente e Servidor.

Resta apenas programar as funções de cada uma das placas NodeMCU, como Cliente e como Servidor e a forma como trocarão mensagens WebSockets. Através da porta micro-USB que as placas possuem é permitido programar através do programa de computador Arduino IDE. Tanto o código do lado do cliente como do servidor encontram-se nos Anexos B e C, respetivamente.

De modo a tornar-se mais facilitada a compreensão do software implementado para o sistema de posição um fluxograma encontra-se representado na página seguinte.

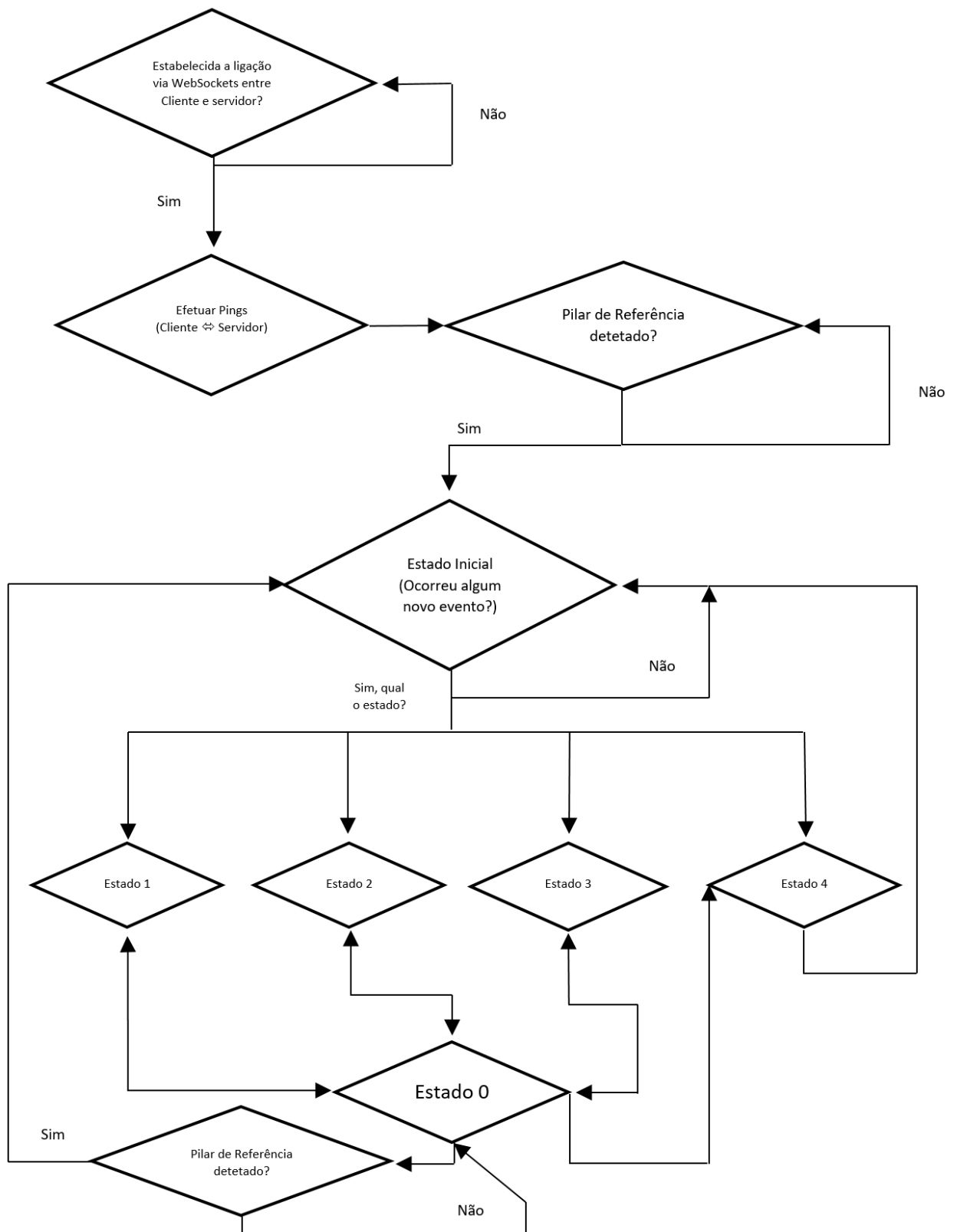


Figura 50. Fluxograma do software implementado.





## 5 TESTES E RESULTADOS

Após a exposição da abordagem adotada para o sistema de localização, neste capítulo retratam-se todos os passos efetuados para o teste do sistema. Na figura 46 encontram-se ilustrados, um esquemático da numeração dos pilares e a localização das duas estruturas introduzidas de modo a permitir a identificação do pilar de referência. No entanto, nada supera um retrato do cenário real do esquemático em questão. Deste modo fotografias da infraestrutura metálica e das estruturas auxiliares foram tiradas desde o local onde se encontram instaladas, o terraço do IT. Como já constatado na figura 42, o carril metálico é suportado por um conjunto de pilares também eles metálicos de uma forma algo característica. Com formato semelhante a um quadrado apenas com as extremidades curvilíneas em vez de efetuarem um ângulo de 90 graus, ao longo dos seus quatro lados encontra-se sempre o mesmo número de pilares (sete) e nas suas quatro curvas encontram-se sempre três pilares. Forma-se assim o número total de 40 pilares ( $28 + 12 = 40$ ). As figuras seguintes ilustram a forma como se implementou o pilar de referência na infraestrutura e dois exemplares de perfis de uma reta e de uma curva do carril metálico (nas figuras 53 e 54 respetivamente).

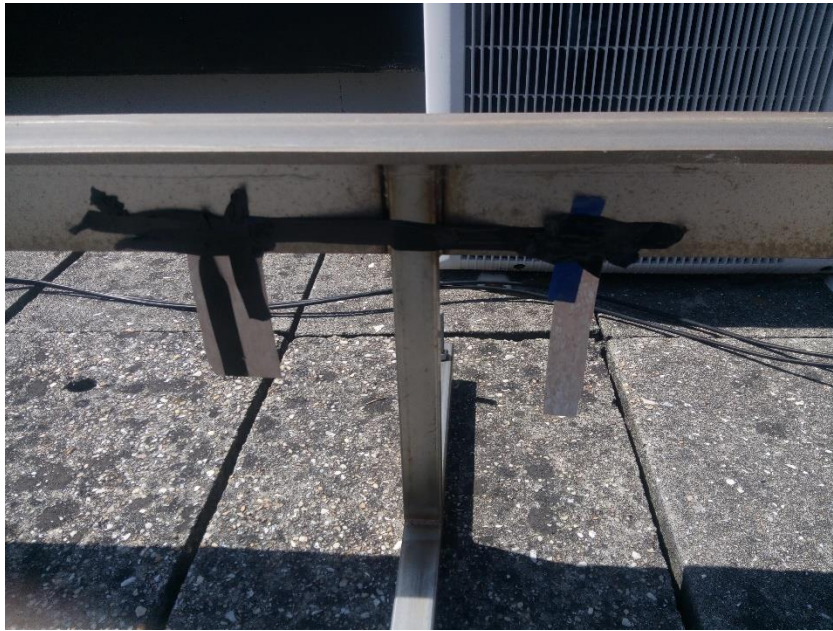


Figura 51. Estruturas colocadas de modo a identificar o pilar de referência.



Figura 52. Pilar referência e restantes pilares vizinhos.



Figura 53. Perfil de uma reta do carril (Pilares número 7 a 13).



Figura 54. Perfil da curva que engloba pilares número 14 a 16.



Estando agora estipulados os números dos pilares e em que local se encontra o pilar de referência, o protótipo ou carga que contém a roda (e respetivo motor) de uma bicicleta elétrica e a caixa de controlo pode ser colocada sobre a infraestrutura metálica.



Figura 55. Montagem do nó móvel sobre o carril metálico.

A figura anterior expõe de que forma o protótipo é colocado sobre a estrutura metálica. Com o auxílio de rodas (a amarelo na figura) o protótipo ou nó móvel é ajustado ao carril de forma a evitar que a estrutura oscile em demasia, conforme a carga vá ganhando velocidade e também para que as curvas sejam feitas de forma segura.

Encontrando-se agora a estrutura montada sobre o carril, é de fácil observação o relatado na figura 45. O par de sensores Infravermelhos BreakBeam introduzidos do lado esquerdo do nó móvel será ligado ao pino digital D1 da placa NodeMCU enquanto que o par de sensores BreakBeam localizados à direita do nó ligar-se-á ao pino digital D2. Desta forma torna-se possível considerar que um deslocamento em direção à esquerda de D1 se considera o sentido CCW (*CounterClockWise* – sentido contrário aos ponteiros do relógio) enquanto que movimento com direção à direita de D2 se considera o sentido CW (*ClockWise* – sentido dos ponteiros do relógio). Na figura seguinte demonstra-se a localização e acondicionamento necessário para os dois pares de recetores de sensores introduzidos na estrutura do nó móvel.



Figura 56. Acomodação e localização dos dois pares de recetores dos sensores infravermelhos BreakBeam.

Através da figura, é possível concluir que as duas ranhuras entre a estrutura metálica e as baterias de cada lado da estrutura são ideais para a introdução dos pares de sensores *BreakBeam*. Do outro lado da estrutura em ranhuras semelhantes, encontram-se alinhados em linha reta os transmissores responsáveis pela emissão de um feixe infravermelho o qual será fulcral para todo o sistema.

Na secção 4.2, mais propriamente na figura 48, encontra-se descrito de que forma as ligações entre a board *NodeMCU*, caixa de controlo e sensores *IR BreakBeam* foram efetuadas. Após alguns segundos de operação, a board *NodeMCU* do lado cliente via uma ferramenta designada por *WiFiManager* (disponível em <https://github.com/tzapu/WiFiManager>), cria um AP (*Access Point*) através da qual o utilizador de uma forma simples pode escolher entre uma lista de APs detetados na vizinhança. A interface gráfica do servidor Web simples lançado pela ferramenta *WiFiManager* é exibida na figura seguinte:

## SocketClienteAP

### WiFiManager

Configure WiFi

Configure WiFi (No Scan)

Info

Reset

```
*WM: AutoConnect
*WM: Connecting as wifi client...
*WM: Using last saved values, should be faster
*WM: Connection result:
*WM: 0
*WM: SET AP STA
*WM:
*WM: Configuring access point...
*WM: SocketClienteAP
*WM: password
*WM: AP IP address:
*WM: 192.168.4.1
*WM: HTTP server started
```

Figura 57. Interface gráfica do servidor Web e código resultante na consola no lado do Cliente.

À direita da figura anterior, constata-se que o NodeMCU do lado Cliente se encontra parametrizado como um AP de nome “*SocketClienteAP*” e com senha “*password*”. O servidor Web encontra-se no endereço IP (192.168.4.1) e a sua interface gráfica respetiva encontra-se à direita da figura. Esta interface permite a escolha de um AP vizinho dentro de uma lista que será apresentada após um varrimento das redes vizinhas disponíveis na área (“*Configure WiFi*”). Caso se conheça qual o nome e senha do AP a aceder pode-se escolher a segunda opção, evitando assim a procura de todas as redes na vizinhança por parte do módulo WiFi ESP8266 (“*Configure WiFi (No Scan)*”). A terceira (“*Info*”) e quarta (“*Reset*”) opções permitem obter informações (tais como endereços IP e MAC do AP) e efetuar o reset ao módulo, respetivamente.

<a href="#">Vodafone-D0739A</a>	🔒 100%	*WM: AutoConnect
<a href="#">ServerAP</a>	🔒 80%	*WM: Connecting as wifi client...
<a href="#">Vodafone-8B148B</a>	🔒 74%	*WM: Using last saved values, should be faster
<a href="#">CAFE:CRISTOVAO</a>	🔒 58%	*WM: Connection result:
<a href="#">ZON-4FA0</a>	🔒 26%	*WM: 3
<a href="#">Vodafone-B6ECB9</a>	🔒 26%	*WM: IP Address:
<a href="#">wifi-link</a>	🔒 22%	*WM: 192.168.4.2
<a href="#">NOS Internet Movei_5E9B</a>	🔒 20%	
<a href="#">NOS_WIFI_Fon</a>	18%	

save

[Scan](#)

Figura 58. Lista de APs disponíveis e código resultante se ligação ao AP escolhido tiver sido bem-sucedida.

Caso se escolha a opção “*Configure WiFi*” (Figura 57), após alguns segundos, o utilizador é reencaminhado para uma listagem de diversos APs disponíveis na área abrangente, onde são dispostos de acordo com a intensidade de sinal da rede (em percentagem) de uma forma descendente. Num primeiro

passo, à esquerda da figura 58, o AP criado pelo lado do servidor (“*ServerAP*”) é escolhido e introduzida a sua respetiva senha. O código resultante, caso a ligação seja bem sucedida, encontra-se apresentado à direita e o endereço IP (“192.168.4.2”) respetivo do ESP8266 do cliente na rede criada pelo servidor é apresentado.

Após este procedimento de registo do cliente na rede WiFi criada pelo servidor, a troca de mensagens WebSocket pode ser inicializada.

```
[WebSocket] Connected to URL: /
[WebSocket] Get text: Ligacao estabelecida!
[WebSocket] Get text: Tudo ok lado Cliente!
```

```
BOOTING....PLEASE WAIT FOR 5 SECOND(S)
BOOTING....PLEASE WAIT FOR 4 SECOND(S)
BOOTING....PLEASE WAIT FOR 3 SECOND(S)
BOOTING....PLEASE WAIT FOR 2 SECOND(S)
BOOTING....PLEASE WAIT FOR 1 SECOND(S)
Connected to:ServerAP
With IP: 192.168.4.1
[SockServ_Num 0] Ligado atraves de 192.168.4.2 url: /
```

Figura 59. Troca de mensagens WebSocket entre cliente (esquerda) e servidor (direita) de inicialização.

Inicialmente esperam-se cinco segundos (devido a parâmetros temporais do NodeMCU) de modo a evitar reinicializações indesejadas por parte da board. De seguida, o AP do servidor de nome (“*ServerAP*”) é criado e o seu endereço IP é apresentado (“192.168.4.1”). A WebSocket é então aberta na porta 81 e o endereço IP do cliente na rede ServerAP é apresentado. Do lado esquerdo da figura, pode-se confirmar a receção de mensagens de que a WebSocket foi estabelecida.

Esta ferramenta foi escolhida por permitir que caso se altere o nome do AP do lado do não seja necessário alterar o código desenvolvido para o NodeMCU cliente. Após escolhido o AP servidor de entre a lista e a inicialização da transmissão de mensagens WebSocket, o sistema encontra-se agora pronto a ser utilizado.

Após a WebSocket se encontrar estabelecida entre Cliente e Servidor, procede-se a um conjunto de pings (5 neste exemplo) efetuados entre o servidor e o módulo WiFi ESP8266 do lado do cliente.

```
Pinging ip:
192.168.4.2
Ping(ms):1
Pinging ip:
192.168.4.2
Ping(ms):48
Pinging ip:
192.168.4.2
Ping(ms):40
Pinging ip:
192.168.4.2
Ping(ms):79
Pinging ip:
192.168.4.2
Ping(ms):9
Pings efetuados! Pode-se proceder ao processo de setup.
```

Figura 60. Pings entre Servidor e Cliente.



É possível constatar que os valores dos pings efetuados abrangem valores que estão compreendidos entre um mínimo de 1 ms até ao máximo de 79 ms.

Numa fase inicial o programa entra numa rotina de inicialização (ou de *setup*) onde o nó móvel efetua o deslocamento necessário de modo a detetar-se o pilar de referência permitindo assim uma contagem de pilares correta e uma estimativa correta da posição respetivamente.

```

Num.quebras via D2:1
Num.quebras via D1:1
Num.quebras via D2:2
Num.quebras via D2:3
Num.quebras via D1:2
Num.quebras via D1:3
Num.quebras via D1:4
Num.quebras via D2:4
Num.quebras via D1:5
Num.quebras via D2:5
Num.quebras via D2:6
Num.quebras via D2:7
Num.quebras via D2:8
Tudo levou Reset. Pilar ref. atingido!

```

Figura 61. Número de quebras de feixe (ou seja, pilares ultrapassados pelo protótipo) detetadas pelos dois pares de sensores, até a deteção do pilar de referência.

A figura anterior apresenta um exemplo desta rotina de *setup*, onde se atravessaram 4 pilares ao longo do percurso do nó móvel. Posteriormente, deteta-se uma diferença de três quebras entre o sensor colocado em D2 e o sensor em D1. Isto indica que o pilar de referência foi atingido e tal é informado ao utilizador.

Após deteção do pilar de referência, o programa do lado do servidor entra na máquina de estados e fica à “escuta” de que uma nova situação ocorra (ou seja, que ocorra uma transição para um dos 5 estados discutidos no subcapítulo 4.2).

```

Estado inicial.Waiting for movement...
Tudo levou Reset. Pilar ref. atingido!
Quebra1:0 // Quebra2:0

```

Figura 62. Estado inicial e informação do número de quebras detetadas por cada sensor.

Na página seguinte, a transição entre os diferentes estados do programa é exibida. Mais tarde, excertos de texto retirados da consola são expostos neste documento de modo a corroborar o que se encontra na figura seguinte.

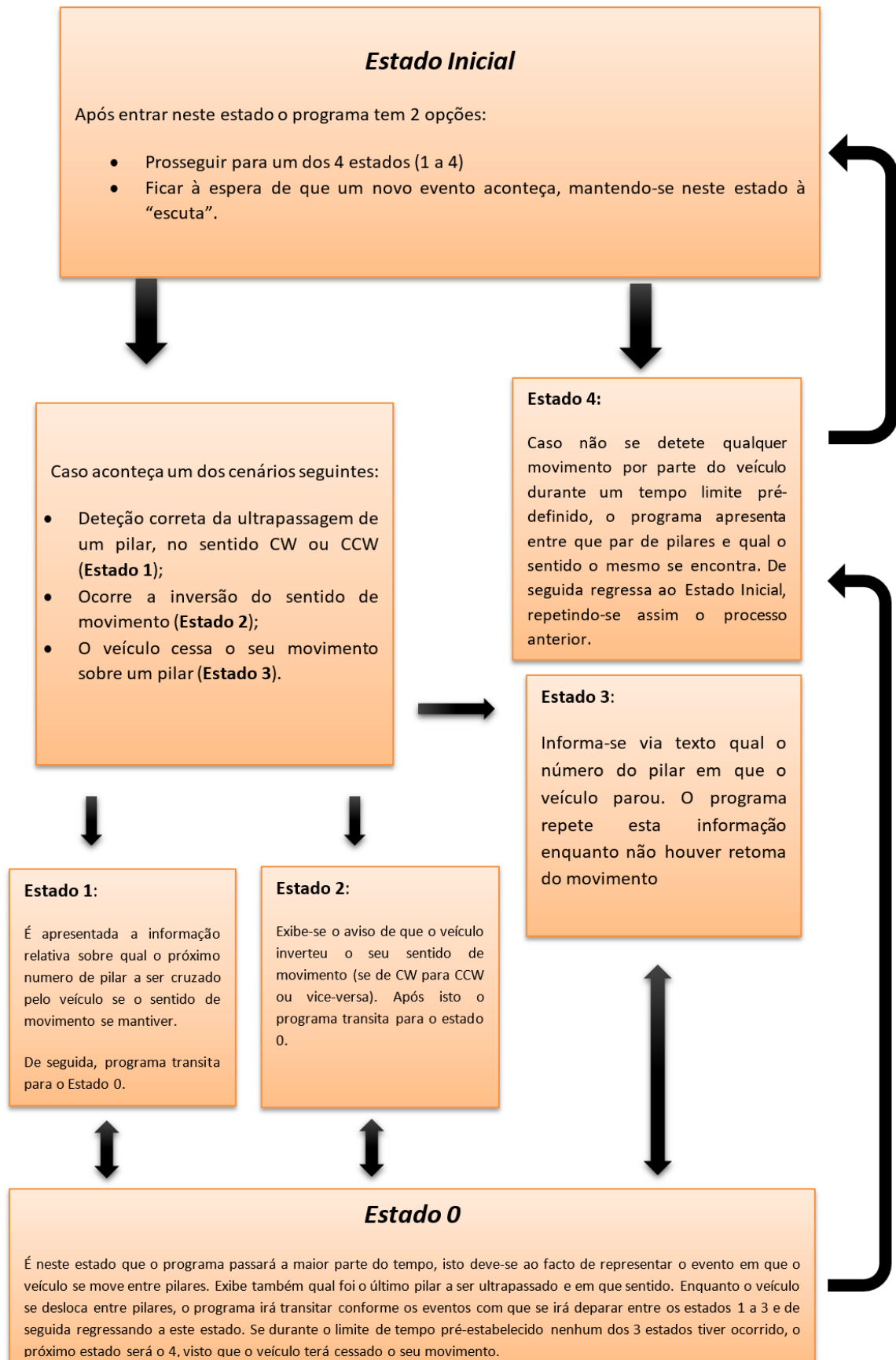


Figura 63. Transições entre os diversos estados do programa.

Diversos excertos de texto foram capturados da consola do programa ArduinoIDE, via comunicação série com um computador do lado do servidor. Estes permitem que a diferenciação nos diversos estados em figuras possa ser efetuada.

```
Quebra1:0 // Quebra2:1
No.Pilar atual:1
No.Pilar seguinte sentido CW:2
Ultimo pilar ultrapassado via CW foi o No.:1
Quebra1:1 // Quebra2:1
```

Figura 64. Estado 0 e estado 1 via CW.

A Figura 64 relata um cenário onde o nó móvel inicia o seu movimento via CW, sendo isto confirmado por inicialmente se detetar uma quebra em D2 e só de seguida em D1. O estado 1 prende-se com a informação relativamente às segunda e terceira linhas do excerto apresentado na figura anterior, onde o programa informa em que pilar se encontra e qual o seguinte caso o sentido se mantenha. O estado 0 recai nas últimas duas linhas, as quais o programa repete com uma taxa de atualização pré-definida de 1s.

```
Quebra1:1 // Quebra2:0
Estado 1 via CCW!
Ultimo pilar a ser cruzado via CCW foi o No.:39
Quebra1:1 // Quebra2:1
```

Figura 65. Estado 0 e estado 1 via CCW.

Cenário em tudo semelhante ao descrito anteriormente, só que agora o movimento iniciou-se no sentido contrário. Inicialmente deteta-se uma quebra em D1 e só depois em D2 indicando assim qual o próximo pilar a ser cruzado caso o sentido de movimento se mantenha. Novamente, estado 1 (segunda e terceira linha da figura) e estado 0 (últimas duas linhas na figura).

```
INVERSAO do sentido em D1! INVERSAO DETETADA via D2!!!
Sensor D1 esta no pilar:39 Sensor D2 esta no pilar:37
```

Figura 66. Estado 2 e Estado 3 (à esquerda D1 e à direita D2).

Estado 2 (relacionado com a situação de inversão no sentido do movimento) e Estado 3 (situação em que um par de sensores fica precisamente sobre um pilar, após se cessar o movimento) encontram-se representados na figura anterior. À esquerda, uma situação em que ocorre inversão de sentido em D1 e posteriormente um cenário em que o sensor em D1 se encontra com o seu feixe quebrado e informação do

pilar em que se encontra (neste exemplo, No.39). Do lado direito da figura, o mesmo cenário é exposto para o par de sensores ligados a D2 (só que neste caso o feixe encontra-se quebrado no pilar número 37).

```
Estado 4!
Carro parou entre pilares 37 e 38 pelo sentido CCW
Estado inicial.Waiting for movement...
```

Figura 67. Estado 4 – Sentido CCW.

Resta agora um retrato do Estado 4. No exemplo anterior encontra-se exposto este estado, o qual informa o utilizador que o carro cessou o seu movimento entre um par de pilares (entre os pilares 37 e 38 neste exemplo) de acordo com o sentido do movimento. De seguida, o programa entra novamente no estado inicial e fica à “escuta” de qual será o estado seguinte.

```
Estado inicial.Waiting for movement...
Estado 4!
Carro parou entre pilares 5 e 6 pelo sentido CW
```

Figura 68. Estado 4 – Sentido CW.

Neste exemplo exibe-se um cenário diferente em que o sentido do movimento e o par de pilares (5 e 6) são distintos.

Concluindo, uma análise crítica ao sistema de localização é retratada. O sistema desenvolvido apresenta algumas limitações começando pela escolha dos sensores infravermelhos. Como é possível verificar na Figura 56 os dois pares de sensores foram cuidadosamente acomodados para que operassem de forma correta quando implementados na estrutura do nó móvel. Isto deve-se aos elevados níveis de luz visível em dias de forte intensidade solar. Operando os sensores à base da radiação infravermelha, erros de leitura ou simplesmente a não leitura correta de um cruzamento de um pilar pode acontecer. No entanto, relativamente a erros passíveis na leitura (situação em que os sensores detetavam mais que uma quebra de feixe, quando deviam apenas detetar uma) conseguem ser corrigidos no código implementado. Excluindo estes problemas (novamente, devido à escolha do tipo de sensores) o sistema efetua uma localização simples, mas correta.



## 6 CONCLUSÕES

O objetivo principal deste documento recaía no desenvolvimento e implementação de um sistema de localização para um nó móvel a ser colocado numa infraestrutura já existente atualmente no terraço do edifício IT em Aveiro. Após todos os testes realizados e resultados obtidos o mesmo pode ser dito que um sistema localização de baixo custo foi implementado com sucesso.

Durante o estudo inicial das diversas redes de testes (*testbeds*) existentes atualmente, foi possível concluir que o aspeto relacionado com aspetos de mobilidade se encontra muitas vezes associado a cenários emulados via computador e raramente em cenários da vida real que possam ser observados e discutidos visualmente permitindo assim uma perceção distinta do que se possa melhorar no futuro. E nas redes de teste em que realmente cenários de mobilidade são suportados requerem que entidades físicas sejam responsáveis pelo transporte dos nós moveis (pessoas carregando portáteis) ou implicam infraestruturas com elevado custo de implementação. A rede de testes AMazING, que se encontra instalada no terraço do edifício IT, constituída por um conjunto de 24 nós fixos ao aliar esta infraestrutura (carril metálico mais protótipo de uma roda de uma bicicleta elétrica como nó móvel) à existente permite que cenários de mobilidade na vida real possam ser observados além de cenários emulados tornando-a assim uma rede de testes ainda mais distinta na comunidade científica.

Posto tudo isto, apenas se pode concluir que a exploração e o desenvolvimento de diferentes técnicas ou abordagens para a implementação de sistemas de posição ou localização de nós munidos de mobilidade levarão ao surgimento de distintas e inovadoras redes de testes no futuro.

## 6.1 TRABALHO FUTURO

Entre os principais aspetos a serem implementados ou melhorados no futuro encontram-se: o tipo de sensores utilizados na abordagem retratada neste documento, a interligação com o sistema de controlo responsável pela velocidade do nó móvel e a implementação de uma interface gráfica que permita ao utilizador visualizar parâmetros relativos à velocidade e posição do nó móvel no carril metálico instalado no terraço do IT. Relativamente à escolha dos sensores a serem utilizados para a estimativa da posição do nó móvel, uma melhoria significativa na precisão do sistema seria obtida caso se optassem por sensores que utilizassem coordenadas GPS (como por exemplo as *boards Tiny RTK* retratadas neste documento). Embora esta abordagem resultasse num custo de implementação algo moderado, a sua preciso centimétrica permitira uma melhora significativa na localização do nó móvel conforme este se deslocasse ao longo do carril metálico.

## BIBLIOGRAFIA

- [1] “Chegada ao Brasil.” [Online]. Disponível em: <http://www.radiopositiva.net/site/2014/04/page/4/>.
- [2] “GNSS.” [Online]. Disponível em: <https://www.meinbergglobal.com/english/news/multi-gnss-support-for-ims-and-lantime-m-series-gps-galileo-glonass-and-beidou.htm>.
- [3] “Segmentos de um GNSS.” [Online]. Disponível em: <https://engrsalmanasghar.wordpress.com/>].
- [4] “Multipath.” [Online]. Disponível em: <http://articles.extension.org/PAGES/9672/GPS-APPLICATIONS-IN-CROP-PRODUCTION>.
- [5] “Ethernert Reader.” [Online]. Disponível em: <https://www.atlasrfidstore.com/impinj-speedway-revolution-r420-uhf-rfid-reader-4-port/>.
- [6] “WiFiReader.” [Online]. Disponível em: <https://www.atlasrfidstore.com/thingmagic-m6-uhf-rfid-reader-4-port-wi-fi>.
- [7] “HandHeld Reader.” [Online]. Disponível em: <https://www.atlasrfidstore.com/tsl-1128-bluetooth-uhf-rfid-reader>.
- [8] “Integrated RFID Reader.” [Online]. Disponível em: <https://www.atlasrfidstore.com/thingmagic-astra-ex-integrated-rfid-reader-wi-fi>.
- [9] “Metal Tag.” [Online]. Disponível em: <https://www.atlasrfidstore.com/omni-id-exo-750-rfid-tag-pack-of-10/>.
- [10] “Wet Tag.” [Online]. Disponível em: <https://www.atlasrfidstore.com/tagsys-lintrak-c-uhf-rfid-tag/>.
- [11] “u-blox M8 high precision GNSS modules datasheet.”
- [12] E. Casanova and S. Quijada, “A New Beacon Navigation System,” *Eis. Uva. Es*, pp. 1–8, 2007.
- [13] “Acelerometros.” [Online]. Disponível em: <http://www.explainthatstuff.com/accelerometers.html>.
- [14] J. Fraden, *HANDBOOK OF MODERN SENSORS, Physics, Design and Applications, Third Edition*. .
- [15] E. Udd, “Fiber optic sensors based on the Sagnac interferometer and passive ring resonator,” *Fiber Opt. Sensors An Introd. Eng. Sci.*, 1991.
- [16] “RLG.” [Online]. Disponível em: <https://www.britannica.com/technology/ring-laser-gyroscope>.
- [17] “Encoder\_incremental.” [Online]. Disponível em: <http://www.ebah.pt/content/ABAAEsEcAJ/encoder-absoluto-incremental>.
- [18] T. F. Andrade, “Encoder Working Principle Theory,” no. February 2016, pp. 1–3.
- [19] J. Borenstein, H. R. Everett, and L. Feng, “Where am I? Sensors and methods for mobile robot positioning,” *Univ. Michigan*, vol. 119, p. 120, 1996.
- [20] E. Micko, “PIR motion sensor,” 2009.

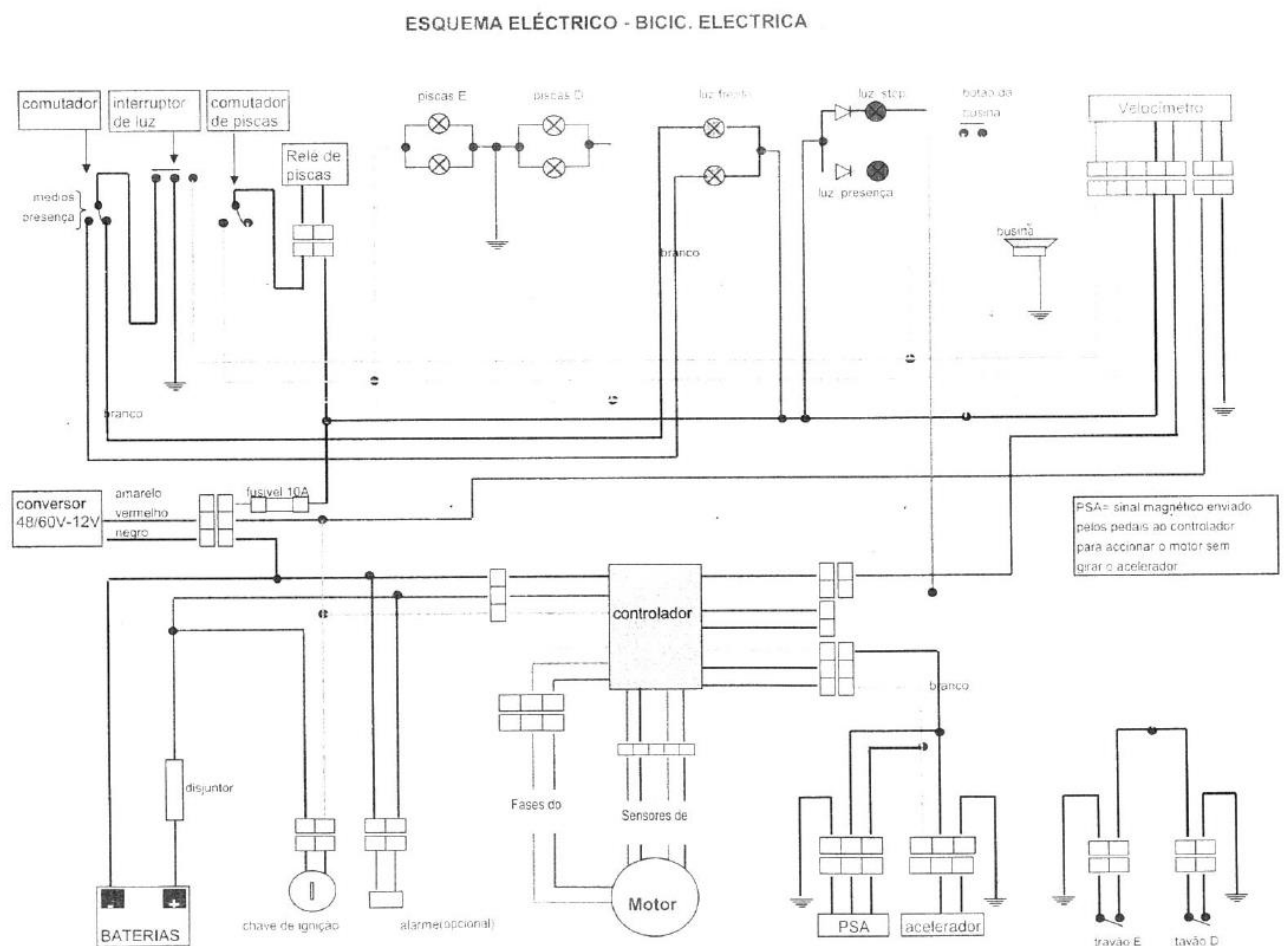


- [21] “IR BreakBeam Sensor.” [Online]. Disponível em: <https://www.adafruit.com/product/2168>.
- [22] “PIR\_adafruit.” [Online]. Disponível em: <https://www.adafruit.com/product/189>.
- [23] C. Schindelhauer, “LNCS 3831 - Mobility in Wireless Networks,” no. January, pp. 100–116, 2006.
- [24] R. ~H. Katz, “Adaption and mobility in wireless information systems,” *IEEE Pers. Commun.*, vol. 1, no. 1, pp. 6–17, 1994.
- [25] “MANET.” [Online]. Disponível em: <https://www.ietf.org/proceedings/55/177.htm>.
- [26] S. Capkun, J.-P. Hubaux, and L. Buttyan, “Mobility {H}elps {S}ecurity in {A}d {H}oc {N}etworks,” *Mobi{HOC} 2003*, p. 11, 2003.
- [27] P. De, a. Raniwala, S. Sharma, and T. C. Chiueh, “Design considerations for a multihop wireless network testbed,” *IEEE Commun. Mag.*, vol. 43, no. 10, pp. 102–109, 2005.
- [28] M. Kropff, T. Krop, M. Hollick, P. Mogre, and R. Steinmetz, “A Survey on Real-World and Emulation Testbeds for Mobile Ad hoc Networks,” *2nd Int. Conf. Testbeds Res. Infrastructures Dev. Networks Communities, 2006. TRIDENTCOM 2006.*, no. Tridentcom, pp. 448–453, 2006.
- [29] “Instituto de Telecomunicações - Aveiro.” [Online]. Disponível em: <https://www.it.pt/ITSites/Index/3>.
- [30] “AMaZING (Advanced Mobile wireless Network playGround).” [Online]. Disponível em: <http://amazing.atnog.av.it.pt/>.
- [31] “Advanced Telecommunications and Networks Group (ATNoG).” [Online]. Disponível em: <http://atnog.av.it.pt/>.
- [32] “Descobrimentos.” [Online]. Disponível em: [https://pt.wikipedia.org/wiki/Era\\_dos\\_Descobrimentos](https://pt.wikipedia.org/wiki/Era_dos_Descobrimentos).
- [33] “Localização.” [Online]. Disponível em: <http://conceito.de/localizacao>.
- [34] Trimble, “The First Global Navigation Satellite System,” *GPS, What It Is How It Work.*, pp. 1–11, 2007.
- [35] “Ethernet, o que é?” [Online]. Disponível em: <http://searchnetworking.techtarget.com/definition/Ethernet>.
- [36] “WiFi Standard.” [Online]. Disponível em: <http://ccm.net/faq/298-what-is-wifi-and-how-does-it-work>.
- [37] “PoE (Power Over Ethernet).” [Online]. Disponível em: <https://www.versatek.com/what-is-poe/>.
- [38] “FCC Standard.” [Online]. Disponível em: <https://www.fcc.gov/general/rules-regulations-title-47>.
- [39] “ETSI Standard.” [Online]. Disponível em: <http://www.etsi.org/technologies-clusters/technologies/radio/rfid>.
- [40] AtlasRFID, “RFID Buyer’s Guide,” 2013.
- [41] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, “Mobile robot positioning: Sensors and techniques,” *J. Robot. Syst.*, vol. 14, no. 4, pp. 231–249, 1997.
- [42] B. Barshan and H. F. Durrant-Whyte, “Inertial navigation systems for mobile robots,” *Robot. Autom. IEEE Trans.*, vol. 11, no. 3, pp. 328–342, 1995.
- [43] RTCM, “RTCM.” [Online]. Disponível em: <http://www.rtcn.org/>.
- [44] “SMA\_Definition.” [Online]. Disponível em: <http://www.pcmag.com/encyclopedia/term/51488/sma-connector>.
- [45] “RTK.” [Online]. Disponível em: <https://drotek.com/en/documentation/tiny-rtk-documentation/>.
- [46] F. O. R. Mobile and R. Everett, *Sensors for Mobile Robots.* .
- [47] L. Gurvits and M. Betke, “Mobile robot localization using landmarks,” *IEEE Trans. Robot. Autom.*, vol. 13, no. no.2, pp. 251–263, 1997.
- [48] S. Thrun, “Bayesian Landmark Learning for Mobile Robot Localization,” vol. 33, no. 1, pp. 41–76, 1998.
- [49] M. J. Mataric, “A Distributed Model for Mobile Robot Environment-Learning and Navigation.” 1990.

- [50] S. Koenig and R. G. Simmons, "Passive Distance Learning for Robot Navigation."
- [51] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," *Perception*, pp. 979–984, 1994.
- [52] S. J. King and C. F. R. Weiman, "HelpMate autonomous mobile robot navigation system," *Proc. SPIE*, vol. 1388. pp. 190–198, 1991.
- [53] H. R. Everett, D. W. Gage, G. A. Gilbreath, R. T. Laird, and R. P. Smurlo, "Real-world issues in warehouse navigation," *Proc. SPIE*, vol. 2352. pp. 249–259, 1995.
- [54] J. Borenstein, "The nursing robot system," PhD Thesis, Technion, Haifa, Israel, 1987.
- [55] W. D. Rencken, "Concurrent localisation and map building for mobile robots using ultrasonic sensors," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, vol. 3, pp. 2192–2197.
- [56] I. J. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. Robot. Autom.*, vol. 7, no. 2, pp. 193–204, Apr. 1991.
- [57] S. B. Thrun, "Exploration and model building in mobile robot domains," in *IEEE International Conference on Neural Networks*, pp. 175–180.
- [58] I. J. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. Robot. Autom.*, vol. 7, no. 2, pp. 193–204, Apr. 1991.
- [59] Dynamics Research Corp., "Techniques For Digitizing Rotary and Linear Motion," pp. 10–11, 1980.
- [60] F. Chenavier and J. L. Crowley, "Position estimation for a mobile robot using vision and odometry," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2588–2593.
- [61] D. S. Nyce, *Linear Position Sensors: Theory and Application*. 2004.
- [62] C. P. De Electrónica and A. Comando, "Sensores e Transdutores – Pt . 1 : Diferença entre Sensores e Transdutores," pp. 3–5.
- [63] "Newton Second Law." [Online]. Disponível em: <https://www.khanacademy.org/science/physics/forces-newtons-laws/newtons-laws-of-motion/a/what-is-newtons-second-law>.
- [64] J. Borenstein, H. R. Everett, and L. Feng, *Where am I? Sensors and methods for mobile robot positioning*. 1996.
- [65] R. E. Meyer, S. Ezekiel, D. W. Stowe, and V. J. Tekippe, "Passive fiber-optic ring resonator for rotation sensing," *Opt. Lett.*, vol. 8, no. 12, p. 644, Dec. 1983.
- [66] S. Nasiri and S. Clara, "A Critical Review of MEMS Gyroscopes Technology and Commercialization Status," *Sensors (Peterborough, NH)*, p. 8, 2009.
- [67] P. Goel, S. Roumeliotis, and G. Sukhatme, "Robot localization using relative and absolute position estimates," *Proc. 1999*, 1999.
- [68] L. E. Navarro-Serment, C. J. J. Paredis, and P. K. Khosla, "A beacon system for the localization of distributed robotic teams," *F. Serv. Robot. Int. Conf.*, vol. 6, pp. 2–7, 1999.
- [69] A. Agent, "The advantages of absolute encoders for motion control," *Sensors*, pp. 19–24, 1991.
- [70] G. Avolio, "Principles of Rotary Optical Encoders," *SENSORS-PETERBOROUGH-*, vol. 10, p. 10, 1993.
- [71] "Astrolabio." [Online]. Disponível em: <http://osdescobridoresbiju.blogspot.pt/p/instrumentos-nauticos.html>.
- [72] "Polarização Circular vs Linear." [Online]. Disponível em: <http://blog.atlasrfidstore.com/circular-polarization-vs-linear-polarization>.



## ANEXO A – ESQUEMA ELÉTRICO DO PROTÓTIPO



Anexo A. Esquema elétrico do protótipo.

## ANEXO B – SOFTWARE (LADO CLIENTE)

```
#include <ESP8266WiFi.h>
#include <WebSocketsClient.h>
#include <WiFiManager.h>

bool estado1, estado2 = false;          // Inicializacao de variaveis responsaveis pela retencao do valor do estado
bool ultimo_estado1, ultimo_estado2 = false; // dos sensores, se devolver HIGH("1") == TRUE, se LOW("0") == FALSE;

const char* ssid= "ESPasAP";
const char* pass = "password1234";

const char* Cont1= "Continuo1";
const char* Queb1 = "Quebra1";

const char* Cont2= "Continuo2";
const char* Queb2 = "Quebra2";

WebSocketsClient socket_cliente; //Socket dedicada ao cliente

void websocketEvent(WStype_t type, uint8_t *payload, size_t length)
{
    switch(type)
    {
        case WStype_DISCONNECTED:
            Serial.printf("[WebSocket] Disconnected!\r\n");
            break;
        case WStype_CONNECTED:
            {
                Serial.printf("[WebSocket] Connected to URL: %s\r\n", payload);
            }
            break;
        case WStype_TEXT:
            Serial.printf("[WebSocket] Get text: %s\r\n", payload);
            break;
        case WStype_BIN:
            Serial.printf("[WebSocket] Get binary length %u\r\n", length);
            hexdump(payload, length);
            break;
        default:
            Serial.printf("[WebSocket] Invalid WStype %d\r\n", type);
            break;
    }
}
```

```

}

}

void setup() {

    Serial.begin(115200); //Começar comunicacao série
    pinMode(D1, INPUT); // Pinos onde se encontram os sensores
    pinMode(D2, INPUT); // definidos como inputs, pois receberao os sinais digitais pretendidos dos sensores
    // pinMode(LED_BUILTIN, OUTPUT); // LED embutido no nodemcu
    // digitalWrite(LED_BUILTIN, HIGH); // esta aqui so pra desligar o LED do nodeMcu

    /*WiFiManager, ferramenta que possibilita uma ligacao facil e rapida a diferentes APs*/
    WiFiManager wifiAP;
    // wifiAP.resetSettings();
    wifiAP.autoConnect("SocketClienteAP","password"); // Criar portal que me leve a ligação pretendida( AP pretendido)
    numa lista que me ira ser mostrada pelo wifimanager

    socket_cliente.begin("192.168.4.1", 81); // Iniciar websocket no IPLocal do ESP8266comoAP atraves da porta 81
    socket_cliente.onEvent(webSocketEvent);
}

void loop() {
    socket_cliente.loop(); // funcao que mantem o socket ligado a espera de troca de msgs
    handle_pins(); // Ler funcao criada para leitura dos pinos e onde se efectua algum processamento para futuro envio da
    info pretendida atraves de sockets
}

void handle_pins(){

    estado1 = digitalRead(D1); // Ler valor digital dos pinos
    estado2 = digitalRead(D2); // D1 e D2

    if (estado1 && !ultimo_estado1) { // Se estado1 for HIGH="1", a leitura digital retorna TRUE.
        Serial.printf("\nFeixe de luz em D1 Continuo \r\n"); // !ultimo_estado1 retorna tb TRUE pois e declarado inicialmente false
        e entra assim neste ciclo "if"
        socket_cliente.sendTXT(Cont1, strlen(Cont1)); // Enviar string "Continuo1" atraves de websocket para o servidor,
        informando-o que tudo esta ok em D1
    }
    if (!estado1 && ultimo_estado1){
        Serial.printf("\nQuebra detetada em D1!!\r\n");
        socket_cliente.sendTXT(Queb1, strlen(Queb1)); // Enviar string "Quebra1" atraves de websockets para o servidor,
        informando-o que houve quebra em D1
    }

    ultimo_estado1 = estado1; // Apenas feito de modo a que a consola nao esteja sempre a imprimir o mesmo resultado,
    mas apenas imprima aquando, unica e exclusivamente, aconteca mudanca de estado

    if(estado2 && !ultimo_estado2){
        Serial.printf("\nFeixe de luz em D2 Continuo\r\n");
        socket_cliente.sendTXT(Cont2, strlen(Cont2));
    }
    if(!estado2 && ultimo_estado2){
        Serial.printf("\nQuebra detetada em D2!!!\r\n");
        socket_cliente.sendTXT(Queb2, strlen(Queb2));
    }

    ultimo_estado2 = estado2;

```

```
}
```

## ANEXO C – SOFTWARE (LADO SERVIDOR)

```
/*    Bibliotecas necessárias à execução do programa    */
```

```
#include <ESP8266WiFi.h>
```

```
#include <WebSocketsServer.h>
```

```
#include <ESP8266Ping.h>
```

```
#include <ESP8266mDNS.h>
```

```
#include <Hash.h>
```

```
#include <user_interface.h>
```

```
/*-----*/
```

```
/*-----Inicialização de variáveis-----*/
```

```
os_timer_t mTimer;
```

```
bool _timeout = false;
```

```
bool state=false;
```

```
bool teste=false;
```

```
bool state1_via_CW = false;
```

```
bool state1_via_CCW=false;
```

```
bool state2_D1=false;
```

```
bool state2_D2=false;
```

```
bool erro=false;
```

```
const int N_total_pilares = 40;
```

```

const long t_threshold = 30000;

/*      Definição das credenciais do AP a ser criado no lado Servidor, ao qual o cliente se irá ligar*/
const char *ssid = "ServerAP";

const char *pass = "asimplepassword";

/*-----*/

bool referencia=false;

bool inicio = true;

bool initial_state=true;


const long t_inv = 5000;      // tempo para deteçao de inversao(5s).


unsigned int quebra1, quebra2, break1, break2, count_erro1,count_erro2=0;

unsigned int d1_state=0, d2_state = 0;

unsigned long d1_cont_t=0, d2_cont_t=0, d1_new_break_t=0, d2_new_break_t=0;

unsigned long previous_break_d1=0, previous_break_d2 = 0;

unsigned int  Num_pilar=0;

unsigned long current_time=0, d1cont_deltat=0, d2cont_deltat=0, d1break_deltat=0, d2break_deltat=0;

signed long t_diff_quebras_CW=0,t_diff_quebras_CCW=0;


/*-----*/

/* Criacao da WebSocket de modo a iniciar-se a comunicacao entre cliente e servidor */


WebSocketsServer websocket = WebSocketsServer(81); // WebSocket via porta 81;

/*-----*/

/*      Função "callback" capaz de detetar que tipo de evento ocorreu      */

void websocketEvent(uint8_t num, WStype_t type, uint8_t* payload, size_t lenght){

    switch(type){

        case WStype_DISCONNECTED:

            Serial.printf("[SockServ_Num %u] Desligado!!\r\n", num);

            break;

```



```

case WType_CONNECTED:

{

    IPAddress ip_socket = websocket.remoteIP(num);

    Serial.printf("[SockServ_Num    %u]    Ligado    atraves    de    %d.%d.%d.%d    url:    %s\r\n",    num,
ip_socket[0],ip_socket[1],ip_socket[2],ip_socket[3], payload);

    websocket.sendTXT(num, "Ligacao estabelecida!",strlen("Ligacao estabelecida!"));

    //De seguida efetuam-se 5 testes ping com origem no IP Servidor e com destino no IP Cliente.

    for(int i=0;i<5;i++)

    {

        const IPAddress teste_ip(192,168,4,2);

        Serial.println("Pinging ip:");

        Serial.println(teste_ip );

        Ping.ping(teste_ip);

        int avg_time_ping=Ping.averageTime();

        Serial.printf("Ping(ms):%d\n",avg_time_ping);

        delay(100);

    }

    Serial.println("Pings efetuados! Pode-se proceder ao processo de setup.");

}

break;

case WType_TEXT:

{

    if(inicio)

        referencia=initial_setup(payload);

    if(referencia){

        inicio=false;

        handle_textValue(payload);

    }

    //websocket.broadcastTXT(payload,lenght);

```

```

    }

    break;

case WStype_BIN:

    Serial.printf("[SockServ_Num: %u] Get binary length %u\r\n",num, lenght);

    hexdump(payload, lenght);

    websocket.sendBIN(num, payload,lenght);

    break;

default:

    Serial.printf("Invalid WStype %d\r\n",type);

    break;

}

}

/*-----*/

/*Função Reset (responsável pela reinicialização dos valores de certas variáveis) /

Executado sempre que se atinge o pilar de referência */

void reset()

{

    Serial.println("Tudo levou Reset. Pilar ref. atingido!");

    d1_new_break_t=d2_new_break_t= 0;

    quebra1=quebra2=Num_pilar=0;

    d1_state=d2_state=HIGH;

    digitalWrite(LED_BUILTIN, HIGH);

}

/*-----*/

/*Função sempre corrida inicialmente para o passo de "setup" de modo a detetar-se

o pilar de referência.

Assim que a diferença entre as quebras do feixe laser dos sensores BreakBeam for

superior a 3, executar rotina chamada reset e o carrinho está perante o pilar de referencia*/

```

```

bool initial_setup(uint8_t* payload)
{
    const char* text = (const char*) payload;

    const char *Queb1 = "Quebra1";
    const char *Queb2 = "Quebra2";


    if(break1>break2)
    {
        if(break1-break2==3)
        {
            reset();

            teste=true;

        }
    }

    if(break2>break1)
    {
        if(break2-break1==3)
        {
            reset();

            teste=true;

        }
    }

    if(strcmp(text,Queb1)==0){

        break1++;

        Serial.printf("Num.quebras via D1:%d\r\n",break1);

    }

    if(strcmp(text,Queb2)==0){

        break2++;
    }
}

```

```

Serial.printf("Num.quebras via D2:%d\r\n",break2);

}

return teste;

}

/*-----*/

/*Função fulcral do programa, responsável pela escolha de estados de acordo com o texto recebido via
Mensagem WebSocket de texto por parte do Cliente*/

void handle_textValue(uint8_t* payload){

/*-----Definição dos valores de certas variáveis importantes para a função-----*/

const long t_upper_limit = 3000; //tempo limite superior para a correta detecção para o estado 1(3s).
const long t_bottom_limit=200; // tempo limite inferior para a correta detecção para o estado 1(200ms).
const char* texto_recebido = (const char*)payload;
const char *Cont1 = "Continuo1";
const char *Queb1 = "Quebra1";
const char *Cont2 = "Continuo2";
const char *Queb2 = "Quebra2";

/*-----*/

if(strcmp(texto_recebido,Cont1)==0)
{
d1_state=HIGH;
d1_cont_t=current_time;
}

if(strcmp(texto_recebido,Queb1)==0)

```

```

{

d1_state=LOW;

previous_break_d1 = d1_cont_t;

d1_new_break_t=current_time;//timestamp de nova quebra em D1

d1break_deltat = d1_new_break_t-previous_break_d1;


/* Correção do erro possível da situação em que haja mais que uma leitura
   por parte do sensor em D1, num intervalo de tempo inferior
   ao estabelecido como limite inferior*/

if(d1break_deltat<t_bottom_limit && !d1_state)

{

if(quebra1>1)

{

count_erro1=count_erro1+1;

Serial.println("ERRO LEITURA EM D1!!!");

erro=true;

}

}

quebra1=(quebra1+1)-count_erro1;

count_erro1=0;

}

/*-----*/

if(strcmp(texto_recebido,Cont2)==0)

{

d2_state=HIGH;

d2_cont_t = current_time;

}


if(strcmp(texto_recebido,Queb2)==0)

{

```

```

d2_state=LOW;

d2_new_break_t=current_time;

previous_break_d2 = d2_cont_t;

d2break_deltat = d2_new_break_t-previous_break_d2;

/* Correção do erro possível da situação em que haja mais que uma leitura
   por parte do sensor em D2, num intervalo de tempo inferior
   ao estabelecido como limite inferior*/

if(d2break_deltat<t_bottom_limit && !d2_state)

{

  if(quebra2>1){

    count_erro2=count_erro2+1;

    Serial.println("ERRO LEITURA EM D2!!");

  }

}

quebra2=(quebra2+1)-count_erro2;

count_erro2=0;

}

/*-----Após correcao efetuada de possiveis leituras erradas, pode-se continuar com o programa!-----*/

/*-----ESTADO 1-----*/

//      Contagem normal dos pilares (no sentido CW como CCW)

/*-----*/

t_diff_quebras_CW = d2_new_break_t - d1_new_break_t;

unsigned long t_diff_quebras_CW_mod = abs(t_diff_quebras_CW);

t_diff_quebras_CCW = d1_new_break_t - d2_new_break_t;

unsigned long t_diff_quebras_CCW_mod = abs(t_diff_quebras_CCW);

if(t_diff_quebras_CW<0)

```

```

{
  if(t_diff_quebras_CW_mod>t_bottom_limit && t_diff_quebras_CW_mod<t_upper_limit && !d1_state)
  {
    //Serial.println("No.Pilar incrementado!Sentido CW");

    if(Num_pilar>=0 && Num_pilar<N_total_pilares)
    {
      Num_pilar++;

      Serial.printf("No.Pilar atual:%d\r\n",Num_pilar);

    }

    if(Num_pilar==N_total_pilares)
    {
      Num_pilar = 0;

      Serial.println("Pilar Referencia atingido!Sentido CW");

    }

    Serial.printf("No.Pilar seguinte sentido CW:%d\r\n",Num_pilar+1);
    state1_via_CW=true; //enviar boolean a true de estado 1 via CW
    t_diff_quebras_CW=0;
  }
}

if(t_diff_quebras_CCW<0){
  if(t_diff_quebras_CCW_mod>t_bottom_limit && t_diff_quebras_CCW_mod<t_upper_limit && !d2_state)
  {
    //Serial.println("No.Pilar decrementado! Sentido CCW");

    if(Num_pilar>0 && Num_pilar<=N_total_pilares){
      Num_pilar--;

      Serial.printf("No.Pilar atual:%d\r\n",Num_pilar);

      if(Num_pilar>0)
        Serial.printf("No.Pilar seguinte no Sentido CCW:%d\r\n",Num_pilar-1);
    }

    if(Num_pilar==0){

```

```

    Num_pilar=N_total_pilares-1;

    Serial.println("Pilar Ref atingido via CCW!!!");

    Serial.printf("No.Pilar seguinte no Sentido CCW:%d\r\n",Num_pilar);

}

state1_via_CCW=true; //enviar boolean a true a informar estado 1 via CCW

t_diff_quebras_CCW=0;

}

}

/*-----ESTADO 2-----*/

    Inversões de sentido tanto de CW para CCW como vice-versa

-----*/

if(d2break_deltat>2000 && d2break_deltat<t_inv && !d2_state && d1_state)
{
    Serial.println("INVERSAO DETETADA via D2!!!");

    state2_D2=true; //Flag da boolean a informar que nos encontramos perante o Estado 2 via D2
}

if(d1break_deltat>2000 && d1break_deltat<t_inv && !d1_state && d2_state)
{
    Serial.println("INVERSAO do sentido em D1!");

    state2_D1=true; // Flag da boolean a informar que nos encontramos perante o Estado 2 via D1
}

/*-----*/

```



```

}

/*-----Final de handle_textValue(payload)-----*/

/*Função callback que coloca a flag de interrupção a "true" */

void tCallback(void *tCall){

    _timeout = true;

}

/*Função que define qual o intervalo de tempo a que se invoca a função callback*/

void usrInit(void){

    os_timer_setfn(&mTimer, tCallback, NULL);

    os_timer_arm(&mTimer, 1000, true); // Estipula-se uma taxa de atualização de 1s.

}

/*-----Função setup (Corpo principal do programa)-----//

* Inicialização de diversos aspetos fundamentais ao programa. */

void setup() {

    Serial.begin(115200);

    pinMode(LED_BUILTIN,OUTPUT); // define-se o LED da board NodeMCU como sinal de saída, para que pisque durante o
Estado 0

    digitalWrite(LED_BUILTIN,HIGH); // Coloca-se inicialmente a "HIGH", pois o LED da board NodeMCU funciona com base em
lógica negativa

    Serial.println();

    delay(2000);

    for(int t=5;t>0;t--){

        Serial.printf("BOOTING....PLEASE WAIT FOR %d SECOND(S)\r\n",t);

        Serial.flush();

        delay(1000);

    }

    usrInit(); //Inicializar função interrupt

    WiFi.softAP(ssid, pass); //Criar AP, com as credenciais definidas no inicio do programa

    Serial.printf("Connected to:%s\r\n",ssid);

```

```

IPAddress ip_esp=WiFi.softAPIP();

Serial.printf("With IP: %d.%d.%d.%d\r\n",ip_esp[0],ip_esp[1],ip_esp[2],ip_esp[3]);


WebSocket.begin();           // Iniciar a WebSocket no endereço IP anterior, via porta 81.

WebSocket.onEvent(WebSocketEvent);    // Iniciar a função callback de eventos descrita anteriormente.

}

/*-----Término da função setup-----*/

/*          Função loop          //

* Corre o código existente no seu corpo em loop infinito */

void loop() {

  WebSocket.loop();    // Manter a WebSocket aberta

  current_time=millis();    // Ler tempo atual

  if(!_timeout){

    if(!initial_state && referencia){

      /*-----ESTADO 3-----//

      *          Estado no qual o carro pára num determinado pilar

      -----*/

      if(!d1_state && d2_state)

      {

        Serial.printf("Sensor D1 esta no pilar:%d\r\n",Num_pilar);

        WebSocket.broadcastTXT("ESTADO 3! via D1",strlen("ESTADO 3! via D1"));

        digitalWrite(LED_BUILTIN,LOW);           // Ligar LED do NodeMCU

      }

      if(!d2_state && d1_state)

```

```

{

Serial.printf("Sensor D2 esta no pilar:%d\r\n",Num_pilar);

websocket.broadcastTXT("ESTADO 3! via D1",strlen("ESTADO 3! via D2"));

digitalWrite(LED_BUILTIN,LOW);                // Ligar LED do NodeMCU

}

////-----//

///<

// *      Informar, via websocket, todos os clientes que se liguem ao servidor
// *      de que nos encontramos no estado 1 ou estado 2
//-----*/

if(state1_via_CW){

Serial.println("ESTADO 1 via CW!");

websocket.broadcastTXT("Estado 1 // Sentido CW",strlen("Estado 1 // Sentido CW"));

state1_via_CW=false;

}

if(state1_via_CCW)

{

Serial.println("Estado 1 via CCW!");

websocket.broadcastTXT("Estado 1 // Sentido CCW",strlen("Estado 1 // Sentido CCW"));

state1_via_CCW=false;

}

if(state2_D1){

Serial.println("Estado 2 via D1!");

websocket.broadcastTXT("Estado 2 via D1",strlen("Estado 2 via D1"));

state2_D1=false;

}

if(state2_D2){

Serial.println("Estado 2 via D2!");

websocket.broadcastTXT("Estado 2 via D2",strlen("Estado 2 via D2"));

```

```

state2_D2=false;

}

/*-----ESTADO 0-----//
*           Carro encontra-se em movimento entre pilares.
-----*/

if(d2_state && d1_state && current_time-d1_new_break_t<t_threshold && current_time-d2_new_break_t<t_threshold){

  websocket.broadcastTXT("Estado 0!",strlen("Estado 0!"));

  if(state){

    state = false;

    //("LED do NodeMCU ON");

  }

  else

  {

    state = true;

    //("LED do NodeMCU OFF");

  }

  digitalWrite(LED_BUILTIN,state); //Colocar o LED da board NodeMCU a "piscar".

  if(t_diff_quebras_CW<0)

    Serial.printf("Ultimo pilar ultrapassado via CW foi o No.:%d\r\n",Num_pilar);

  if(t_diff_quebras_CCW<0)

    Serial.printf("Ultimo pilar a ser cruzado via CCW foi o No.:%d\r\n",Num_pilar);

}

```

```

/*-----ESTADO 4-----*/

* Informar que se ultrapassou o limite de tempo(t_threshold)

* sem que se verificasse algum tipo de movimento!!!

* Informar onde o carro parou!

-----*/

if( (current_time-d1_cont_t>=t_threshold) && (current_time-d2_cont_t>=t_threshold) ){

    Serial.println("Estado 4!");

    if(t_diff_quebras_CCW<0){

        if(Num_pilar==0 || Num_pilar==N_total_pilares)

            Serial.printf("Carro parou entre pilares %d e %d pelo sentido CCW\r\n",N_total_pilares-1,Num_pilar);

        if(Num_pilar >0 && Num_pilar<N_total_pilares)

            Serial.printf("Carro parou entre pilares %d e %d pelo sentido CCW\r\n",Num_pilar-1,Num_pilar);

    }

    if(t_diff_quebras_CW<0)

    {

        if(Num_pilar==0 || Num_pilar == N_total_pilares)

            Serial.printf("Carro parou entre pilares %d e %d pelo sentido CW\r\n",Num_pilar,Num_pilar+1);

        if(Num_pilar>0 && Num_pilar<N_total_pilares)

            Serial.printf("Carro parou entre pilares %d e %d pelo sentido CW\r\n",Num_pilar,Num_pilar+1);

    }

    websocket.broadcastTXT("Estado 4!",strlen("Estado 4!"));

    websocket.broadcastTXT("Carro parou!!",strlen("Carro parou!!"));

    initial_state =true; //Regressar ao estado inicial, definido em baixo.

}

/*-----*/

/* Informar o número de quebras de feixe detetadas em D1 e em D2 */

Serial.printf("Quebra1:%d // Quebra2:%d\r\n",quebra1,quebra2);

if(quebra2>quebra1)

{

```

```

if(quebra2-quebra1==3) // Caso a diferença entre quebras seja igual a 3

    reset();        // Pilar de referencia detetado, logo chamar rotina reset.

}

if(quebra1>quebra2)

{

    if(quebra1-quebra2==3) // Caso a diferença entre quebras seja igual a 3

        reset();        // Pilar de referencia detetado, logo chamar rotina reset.

    }

}

/*-----Estado Inicial-----*/

*   Programa "cai" aqui numa fase inicial ou caso o movimento do carrinho
*   cesse após, um tempo limite estabelecido em 30s.
-----*/

if(initial_state && referencia){

    Serial.println("Estado inicial.Waiting for movement...");

    websocket.broadcastTXT("Estado Inicial!",strlen("Estado Inicial!"));

    websocket.broadcastTXT("Waiting for movement...",strlen("Waiting for movement..."));

    initial_state=false;

}

//-----//

    _timeout=false;

}

yield();

}

/*-----Final da função Loop-----*/

```

